Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 91 (2004) 56–95
www.elsevier.com/locate/entcs

Generating Nearly Sorted Sequences - The use of measures of disorder

Vladimir Estivill-Castro[1]

School of Computing and Information Technology
Griffith University
Brisbane, Australia

Abstract

There have been several formal proposals for a function that evaluates disorder in a sequence. We show here that definitions that allow equivalence to an operational formulation allow for the construction of an algorithm for pseudo-random generation of nearly sorted sequences. As there is interest in comparing performance of algorithms on nearly sorted sequences during experimental evaluations of their implementation, our methods here provide the pathway for establishing the benchmarks datasets to compare such algorithms.

Keywords: Sorting Algorithms, Statistical Correlation of Rankings, Graph Isomorphism, Measures of Disorder, Generation of pseudo-random permutations.

1 Introduction

A measure of disorder estimates the amount of existing disorder in a sequence, and usually gives an approximation to the minimum number of operations of a specific (and sometimes obscure) type required to sort the sequence. For example, the number of inversions in a sequence $X = \langle x_1, \ldots, x_n \rangle$, denoted by $Inv(X)$ and defined by $Inv(X) = \|\{(i,j) \mid i < j, \;\; x_i > x_j\}\|$ (where the items in X belong to some total order and $\|S\|$ denotes the cardinality of a set S). Alternatively, the number of inversions is the minimum number of exchanges of adjacent elements required to sort X or the number of exchanges

[1] Email: v.estivill-castro@griffith.edu.au

1571-0661 © 2004 Elsevier B.V. Open access under CC BY-NC-ND license.
doi:10.1016/j.entcs.2003.12.006

performed by *Bubble Sort*. Interest in measures of disorder is motivated by several applications:

- Measures of disorder formalize the notion that not all sequences of length n require $\Omega(n \log n)$ comparisons to sort them. The adaptive behavior of sorting algorithms is explained with respect to a measure of disorder. For example, Inv has been used to describe the behavior of *Straight Insertion Sort* and has been used in the analysis of several comparison-based sorting algorithms [25,28,34,38,46,47,54]. Work on *Mergesort* by Burge [6] suggested

  > "A measure of the disorder existing in the data is defined as the minimum amount of work required to sort the data into complete order."

  and $Runs$ counts *the number of boundaries between runs*. These boundaries are the so-called "step-downs" [34, page 161], where a smaller element follows a larger one.

  Researchers have tried to demonstrate the practicality of adaptive sorting by testing implementation on nearly sorted sequences [16,39,49,53]. This search for adaptive sorting algorithms that are practical demands the generation of nearly-sorted sequences with a clear understanding if they favor one type of disorder over another. This 'type of disorder' essentially is quantified by the measure of disorders used. Experimental results are usually criticized because the generation of nearly sorted sequences seems to favor the sorting algorithm that is to be used. The approach taken by Levcopoulos and Peterson [39] and Moffat et al [49] is as follows.

  Definition 1.1 Let $N^{<N}$ the set of all finite sequences of natural numbers and let $f, g : N^{<N} \rightarrow \Re$ be two measures of disorder. We say that f is algorithmically finer than g (denoted $f \leq_{alg} g$) if and only if any f-optimal algorithm is also g-optimal [2]. Accordingly, we say f and g are algorithmically equivalent (denoted $f =_{alg} g$) if and only if $f \leq_{alg} g$ and $g \leq_{alg} f$.

- In Statistics, measures of disarray or right invariant metrics are used to obtain coefficients of correlation for rank correlation methods. These coefficients of correlation are used to test the significance of observed rank correlations. For example, Inv appears in the definition of Kendall's τ [31], the most popular coefficient of correlation. However, the distribution of disorder values must be known, and sometimes it is mathematically hard

[2] An algorithm is g-optimal if for all X it sorts X in $O(\|X\| + \log \|\mathit{below}'(g(X), \|X\|, g)\|)$ where $\mathit{below}'(k, n, g)); = \{\pi \in S_n \mid g(\langle \pi \rangle) \leq k\}$, where S_n denotes the group of permutations of $\{1, 2, \ldots, n\}$. Moreover, the identity permutation in S_n is denoted by *id*. The product of two permutations $\pi, \sigma \in S_n$ is denoted by $\pi \cdot \sigma$ and is defined by $\pi \cdot \sigma(i) = \pi(\sigma(i))$. If $\pi \in S_n$, then $\langle \pi \rangle$ denotes the sequence $\langle \pi(1), \pi(2), \ldots, \pi(n) \rangle$.

to obtain a result in this direction. However, pseudo-random generation can obtain tables of values that reflect the properties and parameters of the distribution.

- A benchmark problem arises from the use of graphs in pattern matching. Graphs are combinatorial objects that have been widely used in applications where structured objects emerge in a natural way. Remarkably, in the pattern-matching arena, modeling with graphs has been fruitfully used to match objects [59,61,48]. Thus, the interest in finding efficient algorithms to deal with the Graph Isomorphism (GI) problem, although its precise computational complexity remains unknown [48]. It requires finding a bijection of the vertices so that the edge structure is the same. Labeling the vertices from the same set corresponds to finding a permutation π. However, it is not a minimization problem. Nevertheless, in practice a close variant is a hard minimization problem. In real world applications of pattern matching, the existence of noise, distortion, uncertainty or measurement errors, together with weights associated to nodes and edges, translates the GI problem into its inexact version: the inexact Graph Isomorphism (iGI) or Error-Correcting Graph Isomorphism (ECGI) [59]. To define this problem we first need the notion of attributed graph [59].

  Definition 1.2 [AG] An *attributed graph* is a 4-tuple $G_a = (V, E, \alpha, \beta)$ where $V \neq \emptyset$ is a finite set of vertices; $E \subset V \times V$, is a set of distinct ordered pairs (edges) of distinct elements in V; $\alpha : V \rightarrow \Re$, is a function called the vertex interpreter; and $\beta : E \rightarrow \Re$, is a function called the edge interpreter.

  Definition 1.3 [ECGI] Given two AGs $G_a = (V(G_a), E(G_a), \alpha_G, \beta_G)$ and $H_a = (V(H_a), E(H_a), \alpha_H, \beta_H)$, with $\mid V(G) \mid = \mid V(H) \mid$, the *Error-Correcting Graph Isomorphism* problem is to find a permutation $\pi : V(G_a) \rightarrow V(H_a)$ so that some metric of total dissimilarity between the graph H_a and the graph $G'_a = (\pi[V(G_a)], \pi[E(G_a)], \alpha_{G'}, \beta_{G'})$ is minimized.

  Once again, to evaluate algorithms one needs to generate instances of the error correcting graph isomorphism that are 'easier' because the attributed graphs are not to far apart. Generation with respect to a measure of disorder would typify in what sense the instances are 'easy'.

- The clustering of a data array [44] has two important practical applications: the design of distributed database systems [51] and the design of web sites [62]. These are instances of hard problems (sometimes identified as belonging to the class NP-Hard) where the answer is a permutation (an element in S_n). Links exist, for example, between McCormick *et al.* [44] clustering problem and the Traveling Salesman Problem [35]. It is interest-

> ing then to consider if these problems are solvable in polynomial time if we know that we have a current solution not further away that a parameter k from the optimal solution. In particular, in the context of parameterized complexity [15].

Other applications of measures of disorder include the study of error-sensitivity of sorting algorithms [29,26] and the study of the behavior of external sorting algorithms on nearly sorted sequences [18]. Right invariant metrics have applications in cryptography where they are used to build tests for random permutations [56].

In order for a function to evaluate disorder, its value on a sequence X must depend only on the relative order of the elements in X and not on their particular values. Moreover, it should be minimized when there is no disorder. We formalize these ideas in the following definition.

Definition 1.4 Let $|X|$ denote the length of a sequence X and let $M : N^{<N} \rightarrow \Re$. We say that M is a *measure of disorder*, or *mod*, if,

(i) X sorted implies $M(X) = \min_{|Y|=|X|}\{M(Y)\}$, and

(ii) if $X = \langle x_1, x_2, \ldots, x_n \rangle$, $Y = \langle y_1, y_2, \ldots, y_n \rangle$ and ($x_i \leq x_j$ if and only if $y_i \leq y_j$), for all $i, j \in \{1, 2, \ldots, n\}$, implies $M(X) = M(Y)$.

In this paper we study the set of measures of disorder and how can we obtain operational definitions so we can generate pseudo-random nearly sorted sequences. Section 2 presents right invariant metrics and measures of disorder appearing in the literature. There are many ways to evaluate disorder; however, what constitutes a nearly sorted sequence is intuitively clear.

Mannila observed that Definition 1.4 is too general and, by requiring additional properties, he defined measures of presortedness.

Definition 1.5 Letting $M : N^{<N} \rightarrow N$ be some function, we say that M is a *measure of presortedness* if:

(i) If X is in ascending order, then $M(X) = 0$.

(ii) If $X = \langle x_1, x_2, \ldots, x_n \rangle$, $Y = \langle y_1, y_2, \ldots, y_n \rangle$ and $x_i \leq x_j$ if and only if $y_i \leq y_j$, for all $i, j \in \{1, 2, \ldots, n\}$, then $M(X) = M(Y)$.

(iii) If Y is a subsequence of X, then $M(Y) \leq M(X)$.

(iv) If $X \leq Y$, then $M(XY) \leq M(X) + M(Y)$.

(v) For all x in N, $M(\langle x \rangle X) \leq |X| + M(X)$.

Estivill-Castro, Mannila and Wood examined how well Definition 1.5 reflects intuition and showed that this model was incomplete [17]. Estivill-Castro, Mannila and Wood [17] discussed several important subsets of the set of measures of disorder. Here we show that there are measures of disorder in the literature that do not satisfy Definition 1.5, but for these measures, we can use the proposal by [17] to obtain an equivalent measure of presortedness. However, Definition 1.5 is still unsatisfactory because there are measures of presortedness whose behavior seems to contradict intuition. Despite these problems, Definition 1.5 was a step in the right direction. The tools we use are the necessary and sufficient conditions for a measure of disorder to be extensible to a right invariant metric and the concept of a *normal measure of disorder*. These measures are precisely those that can be used as right invariant metrics. We use the characterization [17] of those *ri-metrics* that are measures of presortedness. These *ri-metrics* are called *regular*.

Here, we summarize results to show that normal measures of disorder, up to ranking, are measures defined in terms of sets of sorting operations. This result is used in Section 5 as the basis for pseudo-randomly generating nearly sorted files.

2 Evaluating disorder

The distance between permutations is particularly useful in statistics for rank correlation methods. The intuition behind it is simple. If we obtain two permutations π, σ in S_n from two independent uniform generators, we expect the distance between π and σ to be close to the average of all possible distances in S_n. However, if the distance between π and σ is small it is reasonable to suspect that there is correlation. Under the hypothesis that the generators are independent, once π has been chosen, σ can be any permutation in S_n with equal probability. A measure of distance from any permutation to the identity is a random variable and its moments (where it is assumed that each permutation has equal probability) are computed. In order to apply this information to π and σ, we must be able to shift π or σ to the identity; that is, we must be able to relabel the data. This brings us to the concept of right invariant metrics.

Diaconis and Graham [11] introduced right invariant metrics (*ri-metrics*) on permutations to evaluate the distance between two permutations. Statisticians normalize these metrics to obtain non-parametric measures of association that have the properties of a rank correlation coefficient [31, page 4]. Kendall's τ, the most popular coefficient of correlation, is defined as $\tau = 1 - 4Inv(\sigma, \pi)/n(n-1)$, where $Inv(\pi, \sigma)$ is the minimum number of pairwise adjacent transpositions required to bring $\langle \pi^{-1} \rangle$ into the order $\langle \sigma^{-1} \rangle$. Fligner and Verducci [21] use *ri-metrics* to generalize Mallow's [41] ranking models. They have studied ranking models based on Cayley's measure and the Hamming distance. Cayley's measure is denoted by Exc, and $Exc(\pi, \sigma)$ is defined as the minimum number of exchanges required to bring $\langle\pi\rangle$ into the order $\langle\sigma\rangle$. The Hamming distance between permutation π and σ is the number of positions where the sequences $\langle\pi\rangle$ and $\langle\sigma\rangle$ differ and it is denoted by $Ham(\pi, \sigma)$.

Definition 2.1 A collection of functions $\{d_n : S_n \times S_n \longrightarrow \Re\}_{n \in N}$ is a *right invariant metric*, or *ri-metric*, if, for all $n \in N$, for all $\pi, \sigma \in S_n$,

(i) $d_n(\pi, \sigma) \geq 0$,

(ii) $d_n(\pi, \sigma) = 0$ if and only if $\pi = \sigma$,

(iii) $d_n(\pi, \sigma) = d_n(\sigma, \pi)$,

(iv) $d_n(\sigma, \pi) \leq d_n(\sigma, \tau) + d_n(\tau, \pi)$, for all $\tau \in S_n$,

(v) $d_n(\sigma, \pi) = d_n(\sigma \cdot \tau, \pi \cdot \tau)$, for all $\tau \in S_n$.

A collection of functions $\{d_n : S_n \times S_n \longrightarrow \Re\}_{n \in N}$ is a *right invariant pseudo-metric* if there is a constant $c > 0$ such that, for all $n \in N$, d_n satisfies 1, 2, 3, and 5 above and

$$d_n(\sigma, \pi) \leq c[d_n(\sigma, \tau) + d_n(\tau, \pi)], \text{ for all } \pi, \sigma, \tau \in S_n.$$

We will omit the subscript of d_n when this is clear from the context. Some examples of *ri-metrics* are immediately obtained from well known vector metrics.

(i) $\|\pi, \sigma\|_p = (\sum_{i=1}^{n} |\pi(i) - \sigma(i)|^p)^{1/p}$, $p \geq 1$. ($p = 1$ is the metric associated with Spearman's footrule [57].)

(ii) $\|\sigma, \pi\|_2 = \sqrt{\sum_{i=1}^{n} |\pi(i) - \sigma(i)|^2}$ is the metric associated with Spearman's coefficient of correlation $\rho = 1 - 6\|\pi, \sigma\|_2^2/(n^3 - n)$.

(iii) $\|\pi, \sigma\|_\infty = \max_{1 \leq i \leq n} |\pi(i) - \sigma(i)|$.

(iv) $M_{01}(\pi, \sigma) = 0$ if $\pi = \sigma$, and $M_{01}(\pi, \sigma) = 1$ otherwise, (the discrete metric).

More *ri-metrics* are obtained from other measures of disarray appearing in the statistical literature. For example, Gordon [22] implicitly defined

$$Grp(\pi, \sigma) = n - \|\{i \mid \pi \cdot \sigma^{-1}(j) < \pi \cdot \sigma^{-1}(k) \text{ whenever } 1 \leq j < i \leq k \leq n\}\|.$$

From the computational point of view, we can use a metric d in $S_{|X|}$ to measure the disorder in X.

Definition 2.2 Given a sequence X of distinct elements from a total order, X defines a permutation $Perm[X]$ in S_n by $Perm[X](i)$ is the final position of x_i when X is sorted. Let d be a metric in $S_{|X|}$, and define M_d by $M_d(X) =$

$d(id, Perm[X])$.

And by definition we obtain:

Lemma 2.3 *If d is a metric, then M_d is a measure of disorder.*

Moreover, some of the *ri-metrics* introduced above give the corresponding measure appearing in the computer science literature. For example, $Inv(X) = Inv(id, Perm[X])$ and this is the number of inversions in X and $Exc(X) = Exc(id, Perm[X])$ is the minimum number of exchanges required to sort X. Because of this, we abbreviate $d(id, \pi)$ by $d(\pi)$.

There have been other motivations for studying different ways to evaluate disorder. Knuth suggested two measures related to *Runs*, namely *Read*, the number of readings[3] in a sequence [34, Sec. 5.1.3 Ex. 20], and *LRuns*, the number of long runs[4] [34, Sec. 5.1.3 Ex. 23]. Burge [6] introduced two measures of presortedness inspired by the number of operations required to merge ascending runs. Let $X = \langle x_1, \ldots, x_n \rangle$ and suppose $Runs(X) = k$ and the $k + 1$ ascending runs have lengths $l_1, l_2, \ldots, l_{k+1}$ with $\sum_{i=1}^{k+1} l_i = n$; then $TWeight1(X)$ is the weight of the lightest binary tree with $k + 1$ leaves and corresponding weights l_i. (The weight of an internal node is the sum of the weights of its two children, and the weight of the tree is the sum of the weights of the internal nodes.) To define the second measure we denote by T_s the minimum weight binary tree with s leaves each of weight 1. Then, $TWeight2(X) = \sum_{i=1}^{d} Weight(T_{\tilde{l}_i})$, where d is the number of runs in $Reverse(X)$ with lengths $\tilde{l}_1, \ldots, \tilde{l}_d$. $Rem(X)$ stands for the smallest number of element that must be removed from X to produce a sorted sequence. Note that $Rem(X) = |X| - Las(X)$ where $Las(X)$ is the length of the largest ascending subsequence of X. This is a very popular measure of disorder, since apparently signals out the items that need to be repositioned. One measure of local disorder is *Dis* that is defined as *the largest distance determined by an inversion.* Related to *Dis* is *Max* defined as *the largest distance an element must travel to reach its sorted position.*

Skiena proposed a measure, named $Enc(X)$, defined as the number of sorted lists constructed by *Melsort* [55] when sorting X. Recall that $\|S\|$ denotes the cardinality of a set S Katajainen, Levcopoulos and Petersson [30,36],

[3] A sequence is said to require k *readings* if we must scan it at least k times from left to right in order to read off its elements in nondecreasing order.

[4] The *long runs* of a sequence are obtained by placing vertical lines just before a segment fails to be monotonic; long runs are either increasing or decreasing, depending on the order of their first two elements, so the length of each long run (except possibly the last) is at least 2.

defined

$$Osc(X) = \sum_{i=1}^{|X|} \|cross(x_i)\| \quad \text{and} \quad Dst(X) = \sum_{i=1}^{|X|} \|rcross(x_i)\|,$$

where $cross(x_i) = \{j \mid 1 \leq j < |X|,\ x_{j+1} < x_i < x_j\}$ and $rcross(x_i) = \{j \mid i < j < |X|,\ x_{j+1} < x_i < x_j\}$. They studied *Heapsort* and *Local Insertion Sort* with respect to these measures. Recently, other measures related to adaptive sorting algorithms have been introduced by Carlsson, Levcopoulos and Petersson [7,37].

The above examples illustrate that disorder can be measured in many ways and although measures differ on the sequences for which they provide small values, what constitutes nearly sorted sequences remains intuitively clear. A sequence is *nearly sorted* if it requires few operations to sort it or it was created from a sorted sequence by a few perturbations. Measures of disorder are a fundamental model but little can be said about them because of their generality. We would like additional properties that bring measures closer to our intuition about disorder. Definition 1.5 captures some important ideas for making measures correspond to our intuition.

3 Consequences of the Axioms

Although the disorder in a sequence can be evaluated in many ways, Mannila [43] proposed the first set of properties that disorder evaluators should satisfy (see Definition 1.5.) We show that if Mannila's properties are used as an axiomatic definition, the resulting mathematical model seems incomplete. Two types of difficulties are discussed. First, there are measures of disorder that do not qualify as measures of presortedness and intuitively they should. Next, we present *EncR*, a function that satisfies Definition 1.5; but it is not (intuitively) a measure of presortedness. This suggests that the formal definition does not correspond entirely to our intuition. Despite these drawbacks, the conditions in Definition 1.5 are sufficient to prove properties and general theorems about measures of presortedness.

3.1 The difficulties with measures of presortedness

Naturally we would like to know if the *ri-metrics* appearing in statistics lead to measures of disorder that qualify as measures of presortedness (Definition 1.5). The first step in this direction is provided by the following result.

Theorem 3.1 *The* ri-metrics $\| \|_\infty$*, Inv, Exc,* M_{01}*, Grp, and Ham lead to measures of disorder that qualify as measures of presortedness.*

Again, this justifies the use of Inv to denote both the measure of presortedness and the *ri-metric*. Also, instead of writing $M_{\| \|_p} = \|id, Perm[X]\|_p$ we simplify the notation and, for a sequence X, we use $\|X\|_p$ to denote $\|id, Perm[X]\|_p$. Note that, for $p \geq 1$, $\| \ \|_p$ does not satisfy axiom 5 in Definition 1.5, since for example, if $n > 1$, $X = \langle 1, 2, \ldots, n \rangle$, and $x = n+1$, then

$$\|\langle x \rangle X\|_p = (n^p + \sum_{i=1}^{n} 1^p)^{1/p} > n = |X| + \|X\|_p.$$

The following result shows how close these metrics are to measures of presortedness. It requires an illustrative but laborious proof (see appendix).

Theorem 3.2 *$\|X\|_p = \|id, Perm[X]\|_p$ satisfies Definition 1.5 except for axiom 5.*

Theorem 3.2 has provided us with an infinite set of measures of disorder; however, these functions do not necessarily qualify as measures of presortedness. Although a function that intuitively evaluates disorder does not qualify as a measure of presortedness, it may be possible to find an equivalent function that qualifies as a measure of presortedness. For example, observe that *Runs* is actually defined as the number of step-downs and not as the number *Read* of ascending runs. This scaling is required so that *Runs* is zero on a sorted sequence (it satisfies axiom 1 in Definition 1.5). Can we use *Read* to estimate disorder? Clearly, it makes sense, *Runs* and *Read* are algorithmically equivalent (see Definition 1.1).

Similarly, Definition 1.5 requires the co-domain of a measure of presortedness to be a set of non-negative integers; however, functions such as $\| \ \|_p$ are real-valued. A real-valued function M satisfying axioms 1 and 2 in Definition 1.5 has a finite image since the domain $S_{|X|}$ is finite; thus, we may use ranking to obtain an algorithmically equivalent measure rk_M defined by

$$rk_M(X) = \|\{M(\langle \pi \rangle) \mid \pi \in S_{|X|} \text{ and } M(\langle \pi \rangle) < M(X)\}\|.$$

The function rk_M scales the measure M to nonnegative integers preserving the property that it evaluates to zero on sorted sequences.

Thus, a measure of presortedness will be a representative of a class of disorder evaluators; namely, all the measures of disorder algorithmically equivalent to it. Although scaling and ranking make Definition 1.5 more flexible, there are other problems.

- There are many measures of disorder in the literature that do not qualify as measures of presortedness for which algorithmically equivalent measures of presortedness are hard to find.

- We would like the space of measures of presortedness to have some natural structure. For example, since disorder can be evaluated in many ways, given two measures M_1 and M_2 of presortedness, $M(X) = M_1(X) + M_2(X)$ is intuitively a measure of presortedness that incorporates both types of evaluation. However, $M(X)$ may fail to qualify as a measure of presortedness. An easy example is to take $M_1 = M_2 = Inv$.
- There are functions that qualify as measures of presortedness but have an unexpected behavior that contradicts intuition.

Let $5a$ be the following axiom: "there is a constant $c > 0$ such that $M(\langle x \rangle X) \leq c|X| + M(X)$". Note that if we replace axiom 5 in Definition 1.5 by axiom $5a$ not only would $\| \|_p$ qualify as a measure of presortedness, for all $p \geq 1$, but we obtain the following result (see Appendix for proof).

Theorem 3.3 *Any linear combination $M = \sum_{i=1}^{r} l_i M_i$, with nonnegative constants l_i, of measures of disorder M_i that satisfy conditions 3 and 4 in Definition 1.5 and axiom* 5a *is a measure of disorder that satisfies conditions 3 and 4 in Definition 1.5 and axiom* 5a.

We now present an example of a function that qualifies as a measure of presortedness but, intuitively, it does not evaluate disorder. Consider the following property:

Prefix Monotonicity: If $X \leq Z$, $Y \leq Z$ and $M(X) \leq M(Y)$, then $M(XZ) \leq M(YZ)$.

We claim that it is intuitively natural to require a measure of presortedness to satisfy this property. To support this claim, we first show that twelve important measures have the prefix monotonicity property.

Theorem 3.4 *The functions Dis, Inv, Rem, Exc, Runs, $\| \|_p$, $\| \|_\infty$, M_{Grp}, Dst, Osc, M_0, M_{01}, satisfy the prefix monotonicity property.*

Note that if $X \leq Y$, then

$$Inv(XY) = Inv(X) + Inv(Y),$$

$$\begin{aligned} Rem(XY) &= Rem(X) + Rem(Y), \\ Exc(XY) &= Exc(X) + Exc(Y), \\ Runs(XY) &= Runs(X) + Runs(Y), \\ \|XY\|_\infty &= \| X\|_\infty + \| Y\|_\infty, \\ M_{Grp}(XY) &= M_{Grp}(X) + M_{Grp}(Y), \end{aligned}$$

and

$$M_0(XY) = M_0(X) + M_0(Y).$$

Thus, for Inv, Rem, Exc, $Runs$, $\| \|_\infty$, M_{Grp} and M_0 the proof follows from the following result.

Lemma 3.5 *Let axiom 4a be "if $X \leq Y$, then $M(XY) = M(X) + M(Y)$." If a function M satisfies axioms 1,2 and 4 in Definition 1.5 and, additionally, axiom 4a, then M satisfies the prefix monotonicity property.*

Proof. Axiom 4 implies that if $X \leq Z$, then $M(XZ) \leq M(X)+M(Z)$; thus, $X \leq Z$, $Y \leq Z$, and $M(X) \leq M(Y)$ imply that $M(XZ) \leq M(X) + M(Z) \leq M(Y) + M(Z) = M(YZ)$. □

To prove Theorem 3.4 for Dis note that

$$\begin{aligned} Dis(XZ) &= \max(Dis(X), Dis(Z)) \\ &\leq \max(Dis(Y), Dis(Z)) = Dis(YZ). \end{aligned}$$

The reader may verify that M_{01} satisfies the monotonicity axiom, but

$$M_{01}(\langle 2,1\rangle\langle 4,3\rangle) < M_{01}(\langle 2,1\rangle) + M_{01}(\langle 4,3\rangle).$$

Therefore, for a monotonic measure of presortedness M, $X \leq Y$ does not necessarily imply that $M(XY) = M(X) + M(Y)$.

The prefix monotonicity property is intuitively desirable. Suppose we replace a prefix with something that has less disorder. Moreover, suppose that all the elements in the prefix and in the replacement prefix are less than every other element in the sequence. If a measure does not satisfy the prefix monotonicity property, then the modified sequence has more disorder than the original sequence. It is displeasing that locally sorting a prefix, that is in sorted order with the remainder of the sequence, introduces disorder.

In order to present an example of a function that qualifies as a measure of presortedness but does not satisfy the monotonicity axiom we describe the measure Enc introduced by Skiena [55]. Given a sequence X, $Enc(X)$ is the number of "dequeues" in the "encroaching set" of X. The encroaching set of a sequence $X = \langle x_1, \ldots, x_n\rangle$ is defined by the following procedure: We say that x_i fits a dequeue D if x_i can be added to the front or the end of D so as to maintain D in sorted order. Repeatedly insert x_i into the first dequeue that it fits. Create a new dequeue if x_i does not fit in any existing dequeue. An example will make this process clear. Consider the sequence $X = \langle 4,6,5,2,9,1,3,8,0,7\rangle$. Initially, D_1 consists of 4, and the second element fits at the end of D_1. 5 is between 4 and 6, so 5 is added to an empty D_2. The next three elements all fit in D_1 and are placed there. 3 does not fit in D_1 but it fits at the front of D_2. Similarly, 8 fits at the end of D_2. 0 fits in

D_1, but the last element requires a new dequeue. The final encroaching set is

$$\{D_1 = [0, 1, 2, 4, 6, 9], \quad D_2 = [3, 5, 8], \quad D_3 = [7]\}.$$

Thus $Enc(\langle 4, 6, 5, 2, 9, 1, 3, 8, 0, 7\rangle) = 3$. Enc does not satisfy Definition 1.5 even if it is scaled. That is, $Enc_{-1}(X) = Enc(X) - 1$ does not satisfy axiom 4, since for $X = \langle 1, 5, 2, 4, 3\rangle$ and $Y = \langle 10, 9, 8, 7, 6\rangle$, $Enc_{-1}(X) = 2$, $Enc_{-1}(Y) = 0$ and $Enc_{-1}(XY) = 3$. However, if we define a new measure

$$M_{Enc}(X) = \begin{cases} 0 & \text{if } X \text{ is sorted} \\ Enc(\langle x_k, \ldots, x_n\rangle) & \text{otherwise, where} \\ & \langle x_1, \ldots, x_{k-1}\rangle \text{ is sorted and } x_{k-1} > x_k \end{cases}$$

we obtain a measure of presortedness that is algorithmically equivalent to Enc.

We are now ready to define our counterexample. For a sequence $X = \langle x_1, x_2, \ldots, x_n\rangle$ define

$$EncR(X) = \begin{cases} 0 & \text{if } X \text{ is sorted} \\ Enc(Reverse(P_X)) & \text{otherwise} \end{cases}$$

where $P_X = \langle x_1, \ldots, x_k\rangle$, $x_{k-1} > x_k$ and $S_X = \langle x_k, \ldots, x_n\rangle$ is sorted. In other words, if X is not sorted, $EncR(X)$ is computed by taking a prefix from X that contains the last step-down in X. If X is sorted, we let $P_X = \langle\rangle$. Then, Enc is computed on the reverse of P_X.

Theorem 3.6 *EncR is a measure of presortedness.*

See appendix for proof. Although $EncR$ is a measure of presortedness, $EncR$ does not satisfy the prefix monotonicity property. Let $X = \langle 5, 4, 3, 2\rangle$, $Y = \langle 3, 4, 5, 2, 1\rangle$, and $Z = \langle 7, 9, 10, 8, 11, 6\rangle$. Thus $EncR(X) = 1$, $EncR(Y) = 2$, and $EncR(Z) = 3$. $EncR(XZ) = 4$ and $X < Z$, but $EncR(YZ) = 3$ and $Y < Z$. Therefore, $EncR$ demonstrates that there are measures of presortedness with displeasing properties.

3.2 Monotonic measures

Mannila [42] was the first to define a measure of presortedness.

> "[the properties of Definition 1.5] are general conditions which any measure of presortedness should satisfy."

But, when discussing future work, he stated that

> "The properties (1)-(5) proposed for a measure are not strong. In trying to

strengthen the results of Section 5, additional properties could be useful."

The following property, which implies the prefix monotonicity property, is intuitively natural.

Definition 3.7 Let $M : N^{<N} \rightarrow \Re$ be a measure of disorder; we say that M is a *monotonic* measure of disorder if $W \leq X \leq Z$, $W \leq Y \leq Z$ and $M(X) \leq M(Y)$ imply that $M(WXZ) \leq M(WYZ)$.

Mannila [42] used two approaches to justify the conditions for a measure of presortedness. He considered a *concrete approach*, where disorder is quantified by the number of operations of a given type which are needed to sort the input, and an *information-theoretic approach*, where disorder is quantified by how much information of the form $x_i < x_j$ must be collected to identify a sequence.

Using the same concrete approach we observe that if WYZ is a sequence where $W < Y$ and $Y < Z$, then the elements in Y are in their correct positions with respect to the elements in W and Z. If we rearrange the elements of Y and obtain X with $M(X) < M(Y)$; then, according to M, there is less disorder in X than in Y. That is, less operations are needed to sort X than to sort Y. Thus, sorting WXZ needs no more operations than sorting WYZ.

Using the information theoretic approach, if $W < X$, $W < Y$, $X < Z$, $Y < Z$ and no more comparisons are needed to identify X than to identify Y, then to identify WXZ requires comparisons to identify W, X and Z. But the number of comparisons needed cannot be more than those required to identify WYZ under $W < Y < Z$.

Note that monotonicity is not implied by the conditions in Definition 1.5, since monotonicity implies the prefix monotonicity property. By verifying Definition 3.7 directly, we can prove the following result.

Corollary 3.8 *Inv, Runs, Rem, Exc, Dis,* m_0, m_{10}, $M_{Enc_{[k,A,D]}}$, M_{Grp}, $\| \|_p$ *and* $\| \|_\infty$ *are monotonic measures of disorder.*

Although the conditions for measures of presortedness are incomplete, we demonstrate that it is a step in the right direction. We claim that measures of presortedness provide a constructive mathematical model and are closer to intuition than measures of disorder. Mannila's goal was to obtain conditions for the existence of M-optimal algorithms for a measure M. We are able to establish some general results on the behavior of measures of presortedness.

Theorem 3.9 *Let M be a measure of presortedness, then the sequences $X = \langle n, 1, \ldots, n-1 \rangle$ and $X = \langle 2, 1, 4, 3, 6, 5, \ldots \rangle$ have at most a linear amount of disorder; that is, $M(X)$ is $O(n)$.*

Proof. Let $X = \langle n, 1, \ldots, n-1\rangle$. Then $M(X) \leq |X| + M(\langle 1, \ldots, n-1\rangle)$ by axiom 1.5-5. From axiom 1.5-1, $M(\langle 1, \ldots, n-1\rangle) = 0$; therefore, $M(X) \leq |X|$. Now, let $X = \langle 2, 1, 4, 3, 6, 5, \ldots\rangle$. By axioms 1.5-2 and 1.5-4,

$$M(X) \leq M(\langle 2, 1\rangle) + M(\langle 4, 3, 6, 5, \ldots\rangle).$$

An inductive argument gives $M(X) \leq |X| M(\langle 2, 1\rangle)/2 = O(|X|)$. □

Theorem 3.10 *For any measure of presortedness M and any sequence X, the value $M(X)$ is $O(|X|^2)$.*

Proof. Using axiom 1.5-5 repeatedly we obtain $M(\langle x_1, \ldots, x_n\rangle) \leq \sum_{i=1}^{|X|} i = O(|X|^2)$. □

The following result shows that, if M is a measure of presortedness such that $M(X) = 0$ implies X is sorted, then $W_n = \{\pi \in S_n \mid rk_M(\langle \pi\rangle) = 1\}$ is always a set of generators of S_n.

Lemma 3.11 *Let M be a measure of presortedness such that $M(X) = 0$ implies X is sorted and let $W_n = \{\pi \in S_n \mid rk_M(\langle \pi\rangle) = 1\}$. If $\pi \in S_n$ is a transposition of adjacent elements, then $\pi \in W_n$.*

Proof. Let $\langle \pi\rangle = \langle 1, \ldots, i-1, i+1, i, i+2, \ldots, n\rangle$. $M(\langle \pi\rangle) \neq 0$ since $\langle \pi\rangle$ is not sorted. $M(\langle \pi\rangle) \geq M(\langle i+1, i\rangle) = M(\langle 2, 1\rangle)$ by axioms 2 and 3, Definition 1.5. Now, $M(\langle \pi\rangle) \leq M(\langle 1, \ldots, i-1\rangle) + M(\langle i+1, i\rangle) + M(\langle i+1, \ldots, n\rangle)$ by axiom 4. Therefore, $M(\langle \pi\rangle) = M(\langle 2, 1\rangle)$. Since, for any unsorted sequence X, $M(X) \geq M(\langle 2, 1\rangle)$ by axioms 2 and 3, we conclude that $rk_M(\langle \pi\rangle) = 1$. □

Conversely, if $\pi \in W_n$ implies that π is a transposition of adjacent elements, then $rk_M = Inv$ and M is algorithmically equivalent to Inv.

4 The relation with ri-metrics

In this section, we revise the relationship between ri-metrics and measures of disorder [17]. This reviews the necessary and sufficient conditions for a measure of disorder to be extended to a ri-metric. If a measure of disorder can be extended to a ri-metric, it is called *normal* [17]. Examples of normal measures are M_{01}, Inv, M_{Grp}, Exc, Ham, and Rem. We also recall the necessary and sufficient conditions for a ri-metric to be used as a measure of presortedness. This results allow a method to naturally construct normal measures; namely, providing sets of sorting transformations. Conversely, if we are given a normal measure, we can almost always identify a set of operations that defines the measure up to ranking.

The necessary and sufficient conditions for a measure to be extended to a *ri-metric* are a consequence of the following two technical results [17].

Lemma 4.1 *If d is a ri-metric, then $d(id,\pi) = d(id,\pi^{-1})$; and $d(id,\pi\cdot\sigma) \leq d(id,\pi) + d(id,\sigma)$.*

Recall that if $\pi \in S_n$, then $\langle\pi\rangle$ denotes the sequence $\langle\pi(1),\ldots,\pi(n)\rangle$. We abbreviate $d(id,\pi)$ by $d(\pi)$.

Lemma 4.2 *If M is a measure of disorder such that*

(i) *$M(X) = 0$ implies X is sorted, and*

(ii) *there are constants $a, b \geq 0$, such that, for all $n \in N$ and for all $\pi, \sigma \in S_n$, we have $M(\langle\pi\cdot\sigma\rangle) \leq a\, M(\langle\pi\rangle) + b\, M(\langle\sigma\rangle)$,*

then

$$d_M(\pi,\sigma) = \frac{M(\langle\pi\cdot\sigma^{-1}\rangle) + M(\langle\sigma\cdot\pi^{-1}\rangle)}{a+b}$$

is a ri-pseudo-metric.

In the next definition we describe those measures that are extensible to ri-metrics.

Definition 4.3 Let M be a measure of disorder. We say that M is *normal* if,

(i) $M(X) = 0$ implies X is sorted,

(ii) for all $n \in N$, and for all $\pi \in S_n$, $M(\langle\pi\rangle) = M(\langle\pi^{-1}\rangle)$, and

(iii) for all $n \in N$, and for all $\pi, \sigma \in S_n$, $M(\langle\sigma\cdot\pi\rangle) \leq M(\langle\sigma\rangle) + M(\langle\pi\rangle)$.

Normal measures are well-behaved measures in the following sense. If we are told that there is no disorder in a sequence, then it is because the sequence is sorted. By applying a permutation σ to a sorted sequence and then applying another permutation τ, we can produce only as much disorder as the disorder produced by each of the permutations σ and τ. Since we need apply only π^{-1} to sort a permutation π, and we need apply only π to sort π^{-1}, the disorder in π should be the same as the disorder in π^{-1}.

Theorem 4.4 *Let M be a* mod. *Let $d_M(\pi,\sigma) = (M(\langle\pi\cdot\sigma^{-1}\rangle) + M(\langle\sigma\cdot\pi^{-1}\rangle))/2$. The function d_M is a ri-metric such that $d_M(id,\pi) = M(\langle\pi\rangle)$ if and only if M is normal.*

Examples of this result are *Exc*, M_{Grp}, *Ham*, *Inv*, M_{01}, and *Rem* [17]. Moreover, we have $d_{Rem}(id,\sigma) = Rem(\langle\sigma\rangle)$ and this corresponds to a ri-metric implicitly defined by Gordon [23].

Theorem 4.5 *Any linear combination, with positive coefficients, of normal measures is a normal measure.*

Proof. Let M_i be normal measures of disorder for $i = 1, \ldots, k$, and $c_1, c_2, \ldots, c_k$ be positive constants. We will prove that $M(X) = \sum_{i=1}^{k} c_i M_i(X)$ is a normal *mod*. First, if X is sorted; then, $M_i(X) = 0$, for $i = 1, \ldots, k$; thus $M(X) = 0$. Now, if $M(X) = 0$, since $c_1 > 0$, $M_1(X) = 0$; thus, X is sorted. Second, $M(\langle\pi\rangle) = \sum_{i=1}^{k} c_i M_i(\langle\pi\rangle) = \sum_{i=1}^{k} c_i M_i(\langle\pi^{-1}\rangle) = M(\langle\pi^{-1}\rangle)$. Finally,

$$\begin{aligned} M(\langle\pi\cdot\sigma\rangle) &= \sum_{i=1}^{k} c_i M_i(\langle\pi\cdot\sigma\rangle) \\ &= \sum_{i=1}^{k} c_i \left[M_i(\langle\pi\rangle) + M_i(\langle\sigma\rangle)\right] \\ &= M(\langle\pi\rangle) + M(\langle\sigma\rangle) \end{aligned}$$

as required □

Applying Lemma 3.11 to normal measures gives the following corollary.

Corollary 4.6 *Let M be a normal measure of presortedness and let*

$$W_n = \{\pi \in S_n \mid rk_M(\langle\pi\rangle) = 1\}.$$

If $\pi \in S_n$ is a transposition of adjacent elements, then $\pi \in W_n$.

In fact, for normal measures of presortedness, we obtain the following stronger result than Theorem 3.10.

Theorem 4.7 *If M is a normal measure of presortedness, then there is a $K > 0$ such that, for all $X \in N^{<N}$, $M(X) \leq K \cdot Inv(X)$.*

Proof. By Corollary 4.6, write $Perm[X]$ as the product of $Inv(X)$ transpositions of adjacent elements. Then $M(X) \leq M(\langle 2, 1\rangle)Inv(X)$ as required. □

This implies, for example, that any sorting algorithm that is adaptive with respect to M (the smaller the value of M, the less time is spent by the sorting algorithm) is also adaptive with respect to Inv. Although Inv plays an important role among ri-metrics and measures of presortedness, its role is not yet fully understood.

5 Random generation of nearly sorted sequences

Imagine you would like to test the ability of a Rubik Cube sorter to adapt the number of operations performed to the difficulty of the input configuration.

In fact, you want to test the sorter in different configurations that are no more than k turns away from *Start* (the sorted position) and see if the sorter performs a number of operations proportional to k. How would you shuffle the cube randomly, so that you have some control of the disorder and the distribution of the generated configurations?

We propose the following strategy. Given the cube at *Start*, for $i = 1$ to k, choose with equal probability either to make one of the allowable turns in the cube or not to make a turn. That is, if there are r allowable turns, then the probability of not making a turn is $1/(1+r)$. Clearly, the result is not more than k turns away from *Start* and, in the case $k = 1$, it produces a configuration of the cube that is less than one turn away from *Start* with equal probability.

In this section we show that this strategy is applicable to normal measures. For any normal measure M there are sets of sorting operations $W_n = \{\pi \in S_n \mid rk_M(\langle\pi\rangle) = 1\}$ such that the minimum number of sorting operations in $W_{|X|}$ required to sort X equals the minimum number of sorting operations in $W_{|X|}$ required to construct X from the sorted sequence containing the elements in X [17]; that is, $M_W(X) = M^W(X)$, where

$$M_W(X) = \min\{k \mid \pi_1, \ldots, \pi_k \in W_{|X|} \text{ and } (\ldots(X_{\pi_1})_{\pi_2}\ldots)_{\pi_k} \text{ is sorted}\},$$

and

$$M^W(X) = \min\{k \mid \pi_1, \ldots, \pi_k \in W_{|X|} \text{ and } Perm[X] = \pi_1 \cdot \pi_2 \cdot \ldots \cdot \pi_k\}.$$

In fact, normal measures provide S_n with a regular graph structure.

Definition 5.1 Let M be a normal measure of disorder. We define the *sorting graph* of S_n with respect to M (denoted by $SG_M(n)$) as follows. S_n is the set of vertices of $SG_M(n)$ and two nodes $\pi,\sigma \in S_n$ are adjacent if and only if $d_{M_W}(\pi, \sigma) = 1$. (See Lemma 4.2 and Theorem 4.4).

For example, Knuth [34, Page 13, Figure 1] gives $SG_{Inv}(4)$.

Theorem 5.2 *For any $n \in N$ and any normal measure M, $SG_M(n)$ is a regular graph of degree $\|W_n\| = \|\{\pi \in S_n \mid rk_M(\langle\pi\rangle) = 1\}\| = \|\{\pi \in S_n \mid M_W(\langle\pi\rangle) = 1\}\|$.*

Proof. Let $\sigma \in S_n$ be any vertex in $SG_M(n)$, $\Gamma(\sigma)$ be the set of vertices adjacent to σ, and $W_n = \{\pi \in S_n \mid M_W(\langle\pi\rangle) = 1\}$. We define a bijection ψ from W_n onto $\Gamma(\sigma)$ by $\psi(\tau) = \tau \cdot \sigma$.

Now, since M is normal, d_{M_W} is a *ri-metric* and by Theorem 4.4, $d_{M_W}(\tau \cdot \sigma, \sigma) = d_{M_W}(\tau, id) = M_W(\langle\tau\rangle) = 1$; thus, $\psi(\tau) = \tau \cdot \sigma \in \Gamma(\sigma)$.

Suppose $\tau_1, \tau_2 \in W_n$ and $\psi(\tau_1) = \psi(\tau_2)$. Then, $\tau_1 \cdot \sigma = \tau_2 \cdot \sigma$ or $\tau_1 = \tau_2$ and we conclude that ψ is injective.

To prove ψ is onto, let $\pi \in \Gamma(\sigma)$; we must exhibit a $\tau \ \in W_n$ such that $\psi(\tau) = \pi$. Let $\tau = \pi \cdot \sigma^{-1}$; then, $\psi(\tau) = (\pi \cdot \sigma^{-1}) \cdot \sigma = \pi$ and $M_W(\langle\tau\rangle) = M_W(\langle\pi \cdot \sigma^{-1}\rangle)$. Since M is normal, M_W is normal and d_{M_W} is a *ri-metric*; therefore, $M_W(\langle\pi \cdot \sigma^{-1}\rangle) = d_{M_W}(id, \pi \cdot \sigma^{-1}) = d_{M_W}(\sigma, \pi) = 1$, since $\pi \in \Gamma(\sigma)$. Thus, $\tau \in W_n$ and the proof is complete. □

Theorem 5.2 proves that for a normal measure we obtain the structure of a "vertex-transitive graph" [2]. Since the set W_n of generators may be redundant, methods to efficiently construct "a strong generating sets" [3] may be very useful.

Let M be a measure of disorder. Informally, we say that a *pseudo-random generator of nearly sorted sequences* (PRG) with respect to M is an algorithm that given nonnegative integers k and n uses a pseudo-random number generator to produce a permutation $\pi \in S_n$ with $M(\langle\pi\rangle) \leq k$. A uniform pseudo-random generator can be constructed as follows. Given k and n, list all the permutations in *below*$'(k, n, M) = \{\pi \in S_n \mid M(\langle\pi\rangle) \leq k\}$ in some canonical order and use a uniform pseudo-random number generator to generate the index of one of them. This algorithm, however, takes exponential time for interesting measures; therefore, it is impractical. Also, *below*$'(k, n, M)$ is not a subgroup, so direct application of the methods in [3] is not possible.

Let M be an integer-valued normal measure of presortedness (if M is not integer-valued, we use rk_M which has the same *below* sets). Let $W_n = \{\pi \in S_n \mid M_W(\langle\pi\rangle) = 1\}$. Any PRG must produce $id \in S_n$ for $k = 0$. For $k = 1$, any PRG must produce a permutation in $W_n \cup \{id\}$; moreover, for $k = 1$, a uniform PRG should choose $\pi \in W_n \cup \{id\}$ with equal probability. We generalize these observations to give a procedure to generate pseudo-random nearly sorted sequences with respect to a normal measure of disorder. We assume we have a procedure *Select*(M, n) that generates a permutation τ from $W_n \cup \{id\}$ randomly and uniformly. Note that this is equivalent to uniformly choosing a permutation in *below*$(1, n, M_W)$ an thus, the implementation of *Select* is a much simpler problem that we will discuss later on. We give the procedure *Generate* in Figure 1. Let k and n be nonnegative integers. We initialize π_0 to be $id \in S_n$. Procedure *Select* chooses k permutations τ_i from $W_n \cup \{id\}$ uniformly; that is, the probability of selecting $\tau_i \in W_n \cup \{id\}$ is $1/(1 + \|W_n\|)$. We then form the product of π_{k-1} with the permutations τ_i to give the permutation π_k. Our method for generating pseudo-random nearly sorted sequences has three fundamental features.

First fundamental feature: *Generate* distribution on *equal*$(z, n, M) = \{\pi \in$

```
procedure $Generate(M, n, k)$;
initialize $\pi \leftarrow id \in S_n$;
for $i := 1$ to $k$ do
begin
    $\tau \leftarrow Select(M, n)$;
    $\pi \leftarrow \pi \cdot \tau$
end
```

Fig. 1. *Generate* returns the product of k permutations τ_i with $M(\langle \tau_i \rangle) \leq 1$.

$S_n \mid M(\langle \pi \rangle) = z\}$ (the *equal* sets with respect to M_W) is uniform.

Second fundamental feature: Every permutation in $below'(k, n, M_W)$, which we recall it is defined as $below'(k, n, M_W) = \{\pi \in S_n \mid M_W(\langle \pi \rangle) \leq k\}$, can be generated.

Third fundamental feature: *Generate* is practical for small k and large n.

The following theorem supports the first two fundamental features. We denote *conditional probability* of event A given event B by $Pr[A|B]$, and we recall that it is defined by $Pr[A|B] = P[AB]/P[B]$ if $P[B] > 0$ and is left undefined if $P[B] = 0$.

Theorem 5.3 *Let M be an integer-valued normal measure of disorder and let k and n be nonnegative integers. If $Pr[Select(M, n) = \tau] = 1/(1 + \|W_n\|)$, for each $\tau \in W_n \cup \{id\}$, then*

(i) *for any nonnegative integer $s \leq k$, let*

$$p_{s,k} = Pr[\ Generate(M, n, k) = \tau \mid M(\langle Generate(M, n, k) \rangle) = s\ ],$$

then

$$p_{s,k} = \begin{cases} 1/\|equal(s, n, M_W)\| & \text{if } M_W(\langle \tau \rangle) = s \\ 0 & \text{otherwise} \end{cases}$$

and

(ii) $Pr[Generate(M, n, k) = \tau] > 0$*, for each* $\tau \in below'(k, n, M_W)$.

To prove the theorem we require two technical lemmas.

Lemma 5.4 *Let n be an integer and M be an integer-valued normal measure of disorder; then, for all $\sigma, \tau \in S_n$ such that $M_W(\langle \sigma \rangle) = s = M_W(\langle \tau \rangle)$,*

(i) $\|\Gamma(\sigma) \cap equal(s - 1, n, M_W)\| = \|\Gamma(\tau) \cap equal(s - 1, n, M_W)\|$,

(ii) $\|\Gamma(\sigma) \cap equal(s, n, M_W)\| = \|\Gamma(\tau) \cap equal(s, n, M_W)\|$, *and*

(iii) $\|\Gamma(\sigma) \cap equal(s + 1, n, M_W)\| = \|\Gamma(\tau) \cap equal(s + 1, n, M_W)\|$.

Proof. By induction on s.
Basis: If $s = 0$, then $M_W(\langle\sigma\rangle) = 0 = M_W(\langle\tau\rangle)$ implies that $\sigma = id = \tau$. Thus, 1, 2 and 3 follow trivially.
Induction step: Let $s \geq 1$. To prove 1 use the induction hypothesis for 2. To prove 2 the induction hypothesis for 3. Now,

$$\bigcup_{i=-1}^{1} [\Gamma(\sigma) \cap \mathit{equal}(s+i, n, M_W)] = \Gamma(\sigma)$$

and

$$\bigcup_{i=-1}^{1} [\Gamma(\tau) \cap \mathit{equal}(s+i, n, M_W)] = \Gamma(\tau).$$

Since both of the unions above are disjoint unions and the graph $SG_M(n)$ is regular, $\|\Gamma(\sigma)\| = \|\Gamma(\tau)\|$. This, together with 1 and 2, implies 3. □

Lemma 5.5 *Let n be an integer and M be an integer-valued normal measure of disorder. For all $\tau, \sigma \in S_n$, such that $M_W(\langle\tau\rangle) = M_W(\langle\sigma\rangle)$ we have*

$$Pr[\mathit{Generate}(M, n, k) = \tau] = Pr[\mathit{Generate}(M, n, k) = \sigma],$$

for all $k \geq 0$.

Proof. We prove the lemma by induction on $r = M_W(\langle\tau\rangle) = M_W(\langle\sigma\rangle)$.
Basis: If $r = 0$, then $M_W(\langle\tau\rangle) = 0M_W(\langle\sigma\rangle)$ implies that $\tau = id = \sigma$ and

$$Pr[\mathit{Generate}(M, n, k) = \tau] = Pr[\mathit{Generate}(M, n, k) = \sigma],$$

for all $k \geq 0$.
Induction step: Let $r \geq 1$ and $\tau, \sigma \in S_n$ be such that $r = M_W(\langle\tau\rangle) = M_W(\langle\sigma\rangle)$. We prove that

$$(1) \qquad Pr[\mathit{Generate}(M, n, k) = \tau] = Pr[\mathit{Generate}(M, n, k) = \sigma],$$

for all $k \geq 0$, from which the claim follows directly.

Let $\{\tau_1^{(<)}, \tau_2^{(<)}, \ldots, \tau_l^{(<)}\} = \mathit{equal}(r-1, n, M) \cap \Gamma(\tau)$; that is, $\{\tau_1^{(<)}, \ldots, \tau_l^{(<)}\}$ are the nodes in $SG_M(n)$ that are adjacent to τ and have less disorder than τ. Similarly, let $\{\tau_1^{(=)}, \ldots, \tau_m^{(=)}\} = \mathit{equal}(r, n, M) \cap \Gamma(\tau)$ be the nodes with equal disorder and finally let $\{\tau_1^{(>)}, \ldots, \tau_p^{(>)}\} = \mathit{equal}(r+1, n, M) \cap \Gamma(\tau)$ be the nodes with greater disorder. Then,

$$\begin{aligned} &Pr[\mathit{Generate}(M, n, k) = \tau] = \\ &\frac{1}{\|\Gamma(\tau) \cap \mathit{equal}(r-1, n, M_W)\|} \sum_{i=1}^{l} Pr[\mathit{Generate}(M, n, k-1) = \tau_i^{(<)}] \\ &+ \frac{1}{\|\Gamma(\tau) \cap \mathit{equal}(r, n, M_W)\|} \sum_{i=1}^{m} Pr[\mathit{Generate}(M, n, k-1) = \tau_i^{(=)}] \end{aligned}$$

$$+\frac{1}{\|\Gamma(\tau)\cap \mathit{equal}(r+1,n,M_W)\|}\sum_{i=1}^{r} Pr[\mathit{Generate}(M,n,k-1)=\tau_i^{(>)}].$$

Similarly, by Lemma 5.4, we have

$$Pr[\mathit{Generate}(M,n,k)=\sigma] =$$
$$\frac{1}{\|\Gamma(\sigma)\cap \mathit{equal}(r-1,n,M_W)\|}\sum_{i=1}^{l} Pr[\mathit{Generate}(M,n,k-1)=\sigma i^{(<)}]$$
$$+\frac{1}{\|\Gamma(\sigma)\cap \mathit{equal}(r,n,M_W)\|}\sum_{i=1}^{m} Pr[\mathit{Generate}(M,n,k-1)=\sigma i^{(=)}]$$
$$+\frac{1}{\|\Gamma(\sigma)\cap \mathit{equal}(r+1,n,M_W)\|}\sum_{i=1}^{r} Pr[\mathit{Generate}(M,n,k-1)=\sigma i^{(>)}]$$

The induction hypothesis implies that

$$Pr[\mathit{Generate}(M,n,k-1)=\tau i^{(<)}] = Pr[\mathit{Generate}(M,n,k-1)=\sigma i^{(<)}],$$
$$Pr[\mathit{Generate}(M,n,k-1)=\tau i^{(=)}] = Pr[\mathit{Generate}(M,n,k-1)=\sigma i^{(=)}]$$

and

$$Pr[\mathit{Generate}(M,n,k-1)=\tau i^{(>)}] = Pr[\mathit{Generate}(M,n,k-1)=\sigma i^{(>)}].$$

Thus, Lemma 5.4 prove Equation (1). □

Proof of Theorem 5.3: To prove the first claim let $k \geq 0$ and $0 \leq s \leq k$. We use the definition of conditional probability to show that if $\tau,\sigma \in \mathit{equal}(s,n,M_W)$, then

$$Pr[\,\mathit{Generate}(M,n,k)=\tau \mid M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s\,]$$
$$= Pr[\,\mathit{Generate}(M,n,k)=\sigma \mid M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s\,],$$

from which the first claim in follows immediately. Using Lemma 5.5 we obtain the following derivation.

$$Pr[\,\mathit{Generate}(M,n,k)=\tau \mid M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s\,] =$$
$$= \frac{Pr[\mathit{Generate}(M,n,k)=\tau \text{ and } M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s]}{Pr[M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s]}$$
$$= \frac{Pr[\mathit{Generate}(M,n,k)=\tau \text{ and } M_W(\langle\tau\rangle)=s]}{Pr[M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s]}$$
$$= \frac{Pr[\mathit{Generate}(M,n,k)=\sigma \text{ and } M_W(\langle\sigma\rangle)=s]}{Pr[M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s]}$$
$$= Pr[\,\mathit{Generate}(M,n,k)=\sigma \mid M_W(\langle \mathit{Generate}(M,n,k)\rangle)=s\,]$$

The second claim follows from the fact that every τ in $\mathit{below}'(n,k,M_W)$ can be factored into the product of no more than k permutations in W_n. Thus, every τ can be represented as the product of k permutations in $W_n \cup \{id\}$ and, thus, $Pr[Generate(M.n, k) = \tau] > 0$. □

Theorem 5.3 proves that *Generate* maximizes the entropy [47, Page 174] of the distributions on all *equal* sets. Therefore, at least on these sets, *Generate* does not disclose information that characterizes the generated permutation. Note that *Generate* does not produce uniformly probable permutations on the *below* sets. The uniformity is restricted to the *equal* sets and that is all. Procedure *Generate* should not be used in simulations to corroborate theoretical results for uniform distributions on other sets that are not *equal* sets. However, we will prove three additional features of procedure *Generate* that in some sense show that *Generate* is the best possible.

First additional feature: For any n, and any normal measure M of presortedness, the distribution of the permutations produced by *Generate* tends to the uniform distribution on S_n when k tends to infinity. This is a natural and expected property.

Second additional feature: *Generate* can be generalized to obtain pseudo-random permutations near a given permutation and not only near the identity. Again, this is natural since the normality of the measure gives S_n a regular structure.

Third additional feature: Any other generator that uses the sorting (shuffling) operations defined by the measure and with the properties listed above is equivalent to *Generate*. Thus, *Generate* is in some sense best possible. This means for example, that we cannot shuffle Rubik's cube (without violating the restrictions imposed by the mechanics of the cube) in any other way and obtain a distribution with the properties claimed.

The stochastic process defined by *Generate* is a Markov chain [52, Chapter 4]. $SG_M(n)$ is finite; thus, we assume we have an enumeration of the nodes in $SG_M(n)$ given by $S_n = \{\pi_1, \pi_2, \ldots, \pi_{n!}\}$. Let P_{ij} denote the probability that the process will, at node π_i, next make a transition to node π_j. Thus, we have $P_{ij} \geq 0$, for $i, j \geq 1$, and

$$\sum_{j=1}^{n!} P_{ij} = 1, \quad \text{for } i = 1, \ldots, n!.$$

Moreover, let $r = \|W_n\|$ be the degree of $SG_M(n)$. Then, $P_{ii} = 1/(1 + r)$, for all i, $P_{ij} = 1/(r + 1)$ for r values of j with $j \neq i$, and $P_{ij} = 0$ for all other values of j.

Let P^s_{ij} denote the s-step transition probability; that is, the probability that the process will move from π_i to π_j in s transitions. Recall that node π_i

```
procedure Gen-Generate($\sigma, M, n, k$);
initialize $\pi \leftarrow \sigma \in S_n$;
for $i := 1$ to $k$ do
begin
    $\tau \leftarrow Select(M, n)$;
    $\pi \leftarrow \pi \cdot \tau$
end
```

Fig. 2. *Gen-Generate* returns the product of σ with k permutations τ_i with $M(\langle \tau_i \rangle) \leq 1$

in a Markov chain is said to have period d if $P^s_{ii} = 0$ whenever s is not divisible by d. Since $P_{ii} = 1/(1+r)$ for all i, all nodes are aperiodic. Furthermore, since we have a finite state Markov chain, all states are recurrent[5] and all recurrent states are positive recurrent[6]. This means that we have an irreducible ergodic Markov chain and we can apply Theorem 4.1 from Ross [52, page 145].

Theorem 5.6 *For an irreducible ergodic Markov chain,* $\lim_{s\to\infty} P^s_{ij}$ *exists and is independent of* i*. Furthermore, letting* $P_j = \lim_{s\to\infty} P^s_{ij}$*, for* $j > 0$*, the unique nonnegative solution of* $P_j = \sum_{i=1}^{n!} P_j P_{ij}$*, for* $j > 0$ *and* $\sum_{j=1}^{n!} P_j = 1$*, is* P_j*.*

We observe that $P_j = 1/n!$, for $j = 1, \ldots, n!$, is a solution to the system defined above and we obtain the following theorem.

Theorem 5.7 *Generate*(M, n, k) *produces a distribution of the permutations that converges to the uniform distribution on* S_n *as* k *goes to infinity.*

In fact, *Generate* can be parametrized to allow generation of permutations close to a permutation σ. Instead of initializing π_0 to $id \in S_n$ we initialize π_0 to σ as shown in Figure 2. The properties outlined in Theorem 5.3 and Theorem 5.7 can be extended for the generalized version of *Generate*.

Theorem 5.8 *Let* M *be an integer-valued normal measure of presortedness, let* k *and* n *be nonnegative integers, and let* $\sigma \in S_n$*. If* $Pr[Select(M, n) = \tau] = 1/(1 + \|W_n\|)$*, for all* $\tau \in W_n \cup \{id\}$*, then*

(i) *For all* τ_1, τ_2 *such that* $d_{M_W}(\sigma, \tau_1) = d_{M_W}(\sigma, \tau_2) \leq k$*,*

$$Pr[Gen\text{-}Generate(\sigma, n, M, k) = \tau_1)] = Pr[Gen\text{-}Generate(\sigma, n, M, k) = \tau_2].$$

[5] A state s is recurrent if the probability that, starting in state i, the process will ever reenter state s is 1.

[6] A recurrent state s is positive recurrent if, starting in s, the expected time until the process returns to state s is finite. Positive recurrent aperiodic states are called ergodic.

(ii) *For all τ such that $d_{M_W}(\tau,\sigma) \leq k$, $Pr[\textit{Gen-Generate}(\sigma,n,M,k)=\tau] > 0$.*

(iii) *For all $\tau \in S_n$, $\lim_{k\to\infty} Pr[\textit{Gen-Generate}(\sigma,n,M,k)=\tau] = 1/n!$*

(iv) *For all τ_1,τ_2 such that $d_{M_W}(\sigma,\tau_1) \leq 1$ and $d_{M_W}(\sigma,\tau_2) \leq 1$*

$$Pr[\textit{Gen-Generate}(\sigma,n,M,k)=\tau_1] = Pr[\textit{Gen-Generate}(\sigma,n,M,k)=\tau_2]$$

Moreover, we prove that *Gen-Generate* is the only Markov chain with the properties described in Theorem 5.8.

Theorem 5.9 *Let M be a integer-valued measure of presortedness, A be any Markov chain describing a random walk with states in S_n, and P_{ij} be the probability of A changing from node π_i to π_j. If A is such that*

(i) *$d_{M_W}(\pi_i,\pi_v) = d_{M_W}(\pi_i,\pi_u)$ implies $P^k_{iv} = P^k_{iu}$, for all k; (that is, A is uniform on the equal sets);*

(ii) *$\lim_{s\to\infty} P^s_{ij} = 1/n!$; (that is, A converges to the uniform distribution);*

(iii) *$\pi_i,\pi_j \in S_n$ and $M_W(\pi_i,\pi_j) = k$ implies $P^s_{i,j} = 0$, for $s < k$; (that is, if two items are at distance k it is impossible to move from one to the other in less that k steps);*

then, there is $p > 0$ such that $P_{ii} = p$, for $i = 1,\ldots,n!$, and if $i \neq j$,

$$P_{ij} = \begin{cases} (1-p)/r & \text{if } \pi_i \text{ is adjacent to } \pi_j \text{ in } SG_M(n) \\ 0 & \text{otherwise} \end{cases}$$

Proof. Let π_u,π_v be adjacent to π_i in $SG_M(n)$; then, $d_{M_W}(\pi_i,\pi_u) = 1 = d_{M_W}(\pi_i,\pi_v)$. Therefore, by hypothesis 1, $P_{iu} = P_{iv}$.

Now, if π_i is not adjacent to π_j, then $d_{M_W}(\pi_i,\pi_j) > 1$; thus, $P_{ij} = 0$. To conclude the proof, observe that A must be ergodic and time reversible, which implies that A is symmetric. □

If we require A to maximize the entropy in the *below*$'(1,n,M)$ sets, then A must be equal to the *Gen-Generate*.

Corollary 5.10 *Let A is as in Theorem 5.9. Suppose for $\pi_i,\pi_j,\pi_u \in S_n$, $d_{M_W}(\pi_i,\pi_u) \leq 1$ and also $d_{M_W}(\pi_j,\pi_u) \leq 1$ implies $P_{iu} = P_{ju}$ (it is uniform at each node), then $P_{ii} = 1/(1+r)$ and $P_{ij} = 1/(1+r)$ if and only if $i \neq j$ and π_i is adjacent to π_j in $SG_M(n)$.*

The rate of convergence to the uniform distribution of Markov chains that represent shuffling processes has recently received attention in the statistical literature [1,13,12]. Theorem 5.7 guaranties convergence. In fact, since

Generate defines a random walk on a finite group[7] (S_n) and, by Theorem 5.3, the corresponding probability distribution is not concentrated on a subgroup[8] or a translate[9] of a subgroup. Therefore, we can apply the results of Aldous and Diaconis [1] to strengthen Theorem 5.7.

Theorem 5.11 *The distribution of permutations produced by Generate converges to the uniform distribution at a geometric rate.*

However, for each measure M of disorder, a more precise description of the rate of convergence to the uniform distribution by the shuffling process defined by *Generate* may be achieved. For example,

(i) for Exc, if k is larger than $\frac{1}{2}n \log n$, then *Generate*(Exc,n,k) is very close to uniform and if k is less than $\frac{1}{2}n \log n$, then *Generate*(Exc,n,k) is very far from uniform [13].

(ii) Let σ_i be a permutation that has at most one nontrivial cycle[10] and this nontrivial cycle is a cyclic shift of $\langle 1, \ldots, i\rangle$; that is, $\sigma_i = (1 \ \ldots i)(i+1)\ldots(n)$. Let $W_n = \{\sigma_i \mid i = 2, \ldots, n\}$. Then, *Generate*$(M_W, n, k)$ is very close to uniform if $k > n \log n$ and very far from uniform if $k < n \log n$ [1].

For many other normal measures, a precise description of the rate of convergence is not known. We have modified the proof of the second example above to show that there is a $c > 0$ such that if $k > cn \log n$, then *Generate*(Rem, n, k) is very close to uniform.

We are left with describing how *Select*(M, n) uniformly chooses a permutation τ such that $M(\langle\tau\rangle) \leq 1$. For Inv, the problem is not difficult. We uniformly choose a number t in $\{1, 2, \ldots, n\}$. If $t = n$ we set $\tau = id$. Otherwise, we set $\tau = (t \ \ t{+}1)$; that is, τ is the permutation that swaps the t-th and $(t + 1)$-th elements. Furthermore, in an implementation of *Generate*, rather than building τ and applying it to π_i we swap the t-th and $(t+1)$-th elements of the array representing π_i to obtain π_{i+1}. Thus, $\pi \leftarrow \pi \cdot$ *Select*(M, n) takes

[7] A *group* $(G, \cdot)$ consists of a set G and a binary operation $\cdot : G \times G \to G$, such that:

(i) the operation $\cdot$ is associative,

(ii) there is an identity element e such that, for all $a \in G$, $a \cdot e = a = e \cdot a$, and

(iii) for all $a \in G$, there is an inverse $a^{-1} \in G$ such that $a \cdot a^{-1} = e = a^{-1} \cdot a$.

[8] A subgroup S of a group $(G, \cdot)$ is a subset of G such that $(S, \cdot)$ is a group.

[9] Let $(G, \cdot)$ be a group and $S \subseteq G$ a subgroup. We say that $Sa = \{s \cdot a | s \in S\}$ and $aS = \{a \cdot a | s \in S\}$ are translates of S.

[10] A cycle $(i_0, i_2, \ldots i_{r-1})$ in a permutation π means $\pi(i_j) = j_{(i+1) mod(r)}$. Every permutation can be represented as a product of its cycles. A cycle is a permutation that leaves all other points fixed.

constant time. A similar strategy gives $\mathit{Select}(Exc, n)$.

For Rem, choose $t \in \{1, 2, \ldots, n\}$ uniformly and then select one of the $n-1$ gaps in $\langle 1, 2, \ldots, t-1, t+1, \ldots, n\rangle$ in which to insert t and produce a sequence with $Rem(X) \leq 1$. Since there are permutations that can be obtained in two different ways, in order for $\mathit{Select}(Rem, n)$ to be uniform, the gaps should be selected as follows.
Case $t = 1$. Choose the first gap with probability $p_1 = \frac{1}{|X|+(|X|-2)(|X|-1)}$. Choose the gap $(2, 3)$ with probability $p_2 = \frac{|X|}{2(|X|+(|X|-2)(|X|-1))}$. For $i = 3, \ldots, n$, choose the gap after i with probability $p_i = \frac{1-p_1-p_2}{|X|-2}$.
Case $t = n$. Choose the last gap with probability $p_{|X|} = \frac{1}{|X|+(|X|-2)(|X|-1)}$. Choose the gap $(|X|, |X|-1)$ with probability $p_{|X|-1} = \frac{|X|}{2(|X|+(|X|-2)(|X|-1))}$. For $i = 1, \ldots, |X|-2$, choose the gap before i with probability $p_i = \frac{1-p_n-p_{n-1}}{|X|-2}$.
Case $1 < t$ and $t < n$. Choose the gap $(t-1, t+1)$ with probability $p_c = \frac{1}{|X|+(|X|-2)(|X|-1)}$. Choose the gap $(t-2, t-1)$ and $(t+1, t+2)$ with probability $p_s = \frac{1}{|X|+(|X|-2)(|X|-1)}$. Choose all other gaps with probability $p_i = \frac{1-p_c-2p_s}{|X|-3}$.

The design of procedure $\mathit{Select}(Max, n)$ demands more effort. Let $E_n = \|\mathit{below}'(n, 1, Max)\|$ denote the number of permutations π in S_n such that $Max(\langle\tau\rangle) \leq 1$. Clearly $E_0 = 0$, $E_1 = 1$ and $E_2 = 2$. Now, consider $\pi \in \mathit{below}'(n, 1, Max)$ with $\pi(1) = 1$. Thus,

$$\begin{aligned} 1 \geq Max(\langle\pi\rangle) &= Max(\langle\pi(1)\rangle\langle\pi(2), \ldots, \pi(n)\rangle) \\ &= Max(\langle\pi(2), \ldots, \pi(n)\rangle). \end{aligned}$$

Therefore, $Perm[\langle\pi(2), \ldots, \pi(n)\rangle] \in \mathit{below}'(n-1, 1, Max)$. If $\pi(1) = 2$, then $\pi(2) = 1$ and $1 \geq Max(\langle\pi\rangle) = Max(\langle\pi(1), \pi(2)\rangle\langle\pi(3), \ldots, \pi(n)\rangle) = \max\{1, Max(\langle\pi(3), \ldots, \pi(n)\rangle)\}$. Thus, $Perm[\langle\pi(3), \ldots, \pi(n)\rangle] \in \mathit{below}'(n-2, 1, Max)$. We conclude that $E_n = E_{n-1} + E_{n-2}$ for $n \geq 3$. It is not hard to see that for $n \geq 1$, $E_n = F_{n+1}$ where F_n is the n-th Fibonacci number. Thus, $E_n = \frac{1}{\sqrt{5}}(\phi^{n+1} - \hat{\phi}^{n+1})$ for $n \geq 1$, and we have proved the following theorem.

Theorem 5.12

$$\lim_{n\to\infty} \frac{\|\{\pi \in S_n \mid Max(\langle\pi\rangle) \leq 1\}\|}{\|\{\pi \in S_{n-1} \mid Max(\langle\pi\rangle) \leq 1\}\|} = \phi,$$

where ϕ is the golden ratio.

Using this result we code $\mathit{Select}(Max, n)$ for selecting uniformly from $\mathit{below}'(1, Max, n)$ as shown in Figure 3. We assume that *random* uniformly generates a real number in $[0, 1)$.

Other methods have been proposed to generate pseudo-random nearly

```
procedure $Select(Max, n)$;
initialize $\pi \leftarrow id \in S_n$; $i \leftarrow 1$;
while $i < n$ do
        if $random \leq 1/\phi$ then
            $\pi \leftarrow \pi \cdot (i \;\; i+1)$; $i \leftarrow i+2$;
        else $i \leftarrow i+1$;
```

Fig. 3. The implementation of *Select*(Max, n) allows generation of permutations with small amount of local disorder.

sorted sequences. To carry out their experiments, Cook and Kim [8] designed an *ad hoc* algorithm to generate sequences X with small $Rem(X)$. Their method has several drawbacks. It only works when $Rem(X)$ is much smaller than $|X|$ and, if successful, the set of elements that must be removed from X to obtain a sorted subsequence is unique. This makes the method biased for sequences with a single large sorted subsequence and a small subsequence of elements out of order. Furthermore, no properties about the distribution corresponding to this method are known; thus, simulation results are difficult to analyze.

The method to generate permutations of order n uniformly [33, pages 139-140] is as follows. Let $L = \langle r_1 < \ldots < r_n \rangle$ be a sorted sequence of n distinct elements. Repeatedly, select an element uniformly from L, remove it from L and place it in the output stream. Oommen and Ng [50] presented a generalization of this method. They consider a control vector $S = [s_1, \ldots, s_n]$ with $\sum_{i=1}^{n} s_i = 1$, $s_i \geq 0$ and a conditional control vector $S|L = [s'_1, \ldots, s'_n]$ as the vector of normalized probabilities of S, where the elements still in L are the only ones with nonzero probability. That is,

$$s'_i = \begin{cases} 0 & \text{if } r_i \notin L \\ s_i / \sum_{r_i \in L} s_i & \text{otherwise.} \end{cases}$$

Their method works as follows. Repeatedly, choose an element r_i from L based on the distribution $S|L$, remove it from L and place it in the output stream. Oommen and Ng use a parameter $\rho \in [0,1]$ to specify the n values in the control vector S. They call ρ the degree of uniformity. When $\rho = 0$, the identity permutation is always generated. When $\rho = 1$, $S = [1/n, \ldots, 1/n]$ and the generated permutation is uniformly distributed. As ρ approaches 0, nearly sorted permutations become more likely.

Oommen and Ng propose two strategies to relate ρ to the control vector.

The geometric progression relates ρ and S by

$$s_i = \begin{cases} \frac{1-\rho}{1-\rho^n} & \text{if } i = 1 \\ \rho s_{i-1} & \text{for } i = 2, \ldots, n. \end{cases}$$

and the arithmetic progression defines s_i by

$$s_i = \frac{\frac{n}{2} + (n - i + 1)(1 - \rho)}{\frac{n}{2}[2n + 1 - \rho(n + 1)]}.$$

Unfortunately, Oommen and Ng consider n to be small when n is 2, 3 and 4, and n to be large when n is between 5 and 15. We are interested in values of n between 10,000 and 1,000,000. We have implemented both the geometric progression method and the arithmetic progression method. For the geometric method, note that, $s_i \to 0$ geometrically as $i \to \infty$. Thus, for $i > 24$, the value of s_i is much smaller, than s_1 and s_2. The number of random bits returned by a call to the pseudo-random generator is insufficient to separate s_i and s_{i+1}, for $i > 24$. Several tricks can be tried to keep the number of calls to the random number generator linear; however, there are still numerical problems involved. We consider this method to be impractical for $n > 24$.

The arithmetic progression suffers from a different problem. Note that,

$$s_i = \frac{n}{n^2 + (1 - \rho)n(n + 1)} + \frac{2(1 - \rho)(n - i + 1)}{n^2 + (1 - \rho)n(n + 1)};$$

thus, s_i depends linearly on i and only on the second term while the denominator is quadratic in n. This means that when n is large the s_i are almost equal; therefore, the method produces permutations almost uniformly for almost all ρ. In fact, because of numerical precision, the implementation of this method cannot be distinguished from an uniform generator.

6 Additional remarks

We have discussed four related concepts: measures of disorder, measures of presortedness, normality, and regularity. We have shown that the most interesting and widely used measures of disorder are normal and regular measures of presortedness or at least they are algorithmically equivalent to such measures. Moreover, we have shown that for such measures it is possible to obtain nearly sorted sequences efficiently. These nearly-sorted sequences are necessary to generate benchmark test-sets for a series of important computational problems beyond sorting. For example, for the Error Correcting Graph Isomorphism.

Moreover, it seems that for problems like the Clustering of a Data Array [44] or the Longest Hamiltonian Path Problem (where the answer is a permutation that minimizes/maximizes a criteria that can be evaluated in polynomial time), if we are given a current solution π and a normal measure of presortedness M plus the additional information that for the optimal solution σ we have $M(\pi, \sigma) \leq k$, then Theorem 4.7 implies that this version of the problem is Fix-Parameter tractable [15] (we simply need to search the space of all permutations at distance k or less from π which has size a polynomial in k and the operators can be identified as W_n).

The connections between *mods* and *ri-metrics* leave several open questions. Researchers have attempted to compare *ri-metrics* by establishing inequalities between them [11]. Note that Inv is defined from a set of operations that are exactly all transpositions of adjacent elements. Lemma 3.11 shows that Inv is the normal measure (*ri-metric*) most sensitive to disorder. The popularity of Kendall's τ is due to the fact that Inv is asymptotically normally distributed with known mean and variance for each n. Our results show that normal measures (*ri-metrics* and coefficients of correlation) can be constructed in a similar way, we only need to provide the sets W_n of sorting operations to obtain the *ri-metric* d_{M_W}.

In order to use these *ri-metrics* in statistical applications, the characteristics of the distributions must be described either analytically or by tabulation of their values. Analytical results can be difficult, as suggested by Ulam's problem (computing the limiting behavior of the expected value of $Rem(X)$). Stanley [58] listed Ulam's problem as one of several open problems in enumerative combinatorics and Knuth [34, Section 5.1.4,Problem 28] ranked the problem as *M47*. In fact, several researchers using many different mathematical techniques have contributed to solve the problem. Baer and Brock [4] computed extensive tables and conjectured that $E[Las(X)] = 2\sqrt{|X|}$; thus, $E[Rem(X)] = |X| - 2\sqrt{|X|}$. McKay [45] extended the values computed by Baer and Brock (up to $|X| = 36$) to $|X| = 75$. Dixon [14] proved that

> '...the probability that the length of the longest monotonic subsequence in a sequence $X_1, X_2, \ldots, X_n$ of independent random variables with a common continuous distribution lies in the range $(e^{-1}n^{1/2}, en^{1/2})$ tends to 1 as $n \rightarrow \infty$.'

This shows that the asymptotic expected value of $Lax(X)$ is $\Theta(|X|^{1/2})$. Del Junco and Steele [10] discuss the problem once more in 1979. By then, Hammersley [27] had proved, via an ingenious use of the planar Poisson process, that $\lim_{|X|\rightarrow\infty} |X|^{1/2} E[Las(X)] = c$ where c is a constant and the convergence is in probability. Hammersley [27] gave bounds on c which were improved by Kingman [32]. Then, Logan and Shepp [40] used calculus of variations, Hilbert transform and Fourier transform to prove $c \geq 2$. Vershik and Kerov[7] [60] used similar methods to prove $c \leq 2$. Dallal and Hartigan [9] have performed

Monte Carlo simulations using 10,000 random permutation with length in the range 20 to 400. Their results showed that there is a simple and reasonably accurate relationship between $|X|$ and the variance of $Rem(X)$. Their empirical evidence indicated that the distribution of $Rem(X)$ is asymptotically normal. Much of the recent work on this has been summarized by Baik, Deift and Johansson [5] who elaborated on the Tracy-Widom distribution and the Gaussian Unitary Ensemble of random matrix theory. In fact, more accurate estimates of the constants involved in the limit for the distribution of $E[Rem(X)]$ are now known. The mathematical effort and the sophistication of the techniques for this problem are remarkable.

Since analytical results on the distribution of measures are hard, it is desirable, at least, to characterize those *ri-metrics* that decompose into a sum of independent uniform distributions or other well known distributions. If we want to test correlation or agreement of more than two rankings (because the objects are ranked independently by boards of judges), the corresponding techniques must be developed [19,20,21]. Random generation with bounded disorder may provide the answer for tabulating values of the distributions.

Requiring that a measure used for describing the behavior of a sorting algorithm qualify as a measure of presortedness may seem restrictive. However, in understanding a measure, it is helpful to describe its distribution over $S_{|X|}$ and its relationship to other measures; it is also helpful to analyze its properties, and in particular, to verify if it is normal or regular.

Finally, the arguments here are somewhat biased in that we do not necessarily consider a sequence in descending order to be sorted.[11] The elements in a sequence in descending order are very far from being randomly shuffled; however, some work must be done to obtain the sequence in ascending order. Although Mannila has already argued that $\langle 2, 1\rangle$ should be considered to be a sequence with disorder, we argue that ascending and descending order cannot be considered equivalent. From the point of view of statistics, a ranking in ascending order cannot be considered the same as a ranking in descending order. From the point of view of sorting, the equivalence of ascending order with descending order implies that the operation of reversing a sequence has no cost. From the theoretical point of view, for any sequence X, we can execute two copies of an M-optimal algorithm in parallel, with inputs X and $Reverse(X)$, terminating as soon as either one terminates. This gives an algorithm that sorts X in ascending or descending order, whichever was easier according to M, with a constant delay factor.

[11] The reader familiar with Kolmogorov Complexity may attempt to define a measure of disorder that takes this into account, however, several technical difficulties must be resolved.

References

[1] Aldous D. and P. Diaconis, Shuffling cards and stopping times. *American Mathematical Monthly*, 93:333–348, 1986.

[2] Babai, L., Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 164–174, New Orleans, LA, 1991. SIGACT: Special Interest Group on Algorithms and Computation Theory, ACM Press.

[3] Babai, L., G. Cooperman, L. Finkelstein, E. Luks, and A. Seress, Fast Monte Carlo algorithms for permutation groups. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 90–100, New Orleans, LA, 1991. SIGACT: Special Interest Group on Algorithms and Computation Theory, ACM Press.

[4] Baer, R. M. and P. Brock, Natural sorting permutation spaces. *Mathematics of Computation*, 22:385–410, 1968.

[5] Baik, J., P. Deift and K. Johansson, On the distribution of the length of the longest increasing subsequence of random permutations. *Journal of the American Mathematical Society* 12:(4) 1119-1178, 1999.

[6] Burge, W. H., Sorting, trees and measures of order. *Information and Control*, 1:181–197, 1958.

[7] Carlsson, S., C. Levcopoulos, and O. Petersson, Sublinear merging and natural merge sort. Technical report, Department of Computer Science, Lund University, 1989.

[8] Cook, C. R. and D. J. Kim, Best sorting algorithms for nearly sorted lists. *Communications of the ACM*, 23:620–624, 1980.

[9] Dallal, G. E. and J. A. Hartigan, Note on a test of monotone association insensitive to outliers. *Journal of the American Statistical Association*, 75(371):722–725, 1980.

[10] Del Junco, A. and M. Steele, Hammersley's law for the Van der Corput sequence:an instance of probability theory for pseudorandom numbers. *The Annals of Probability*, 7(2):267–275, 1979.

[11] Diaconis, P, and R. L. Graham, Spearman's footrule as a measure of disarray. *J. Royal Statistical Soc. Series B*, 32(2):262–268, 1977.

[12] Diaconis, P., R. L. Graham and J. A. Morrison, Asymptotic analysis of a random walk on a hypercube with many dimensions. *Random Structures and Algorithms*, 1(1):51–72, 1990.

[13] Diaconis, P. and M. Shahshahani, Generating a random permutation with random transpositions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57:159–179, 1981.

[14] Dixon, J. D., Monotonic subsequences in random sequences. *Discrete Mathematics*, 12:139–142, 1975.

[15] Downey, R. G. and M. R. Fellows, *Parameterized Complexity.* Monographs in Computer Science. Springer-Verlag, 1999.

[16] Elmasry, A., Three sorting algorithms using priority queues. In *Proceedings of 14th International Symposium on Algorithms and Computation (ISAAC 2003)*, Kyoto, Japan, 2004. Springer-Verlag Lecture Notes in Computer Science. to appear.

[17] Estivill-Castro, V., H. Mannila and D. Wood, Right invariant metrics and measures of presortedness. *Discrete Applied Mathematics*, 42:1–16, 1993.

[18] Estivill-Castro, V. and D. Wood, The performance of replacement selection on nearly sorted sequences. Technical Report in preparation, Department of Computer Science, University of Waterloo, 1991.

[19] Feigin, P. D. and M. Alvo, Intergroup diversity on concordance for ranking data: An approach via metrics for permutations. *The Annals of Statistics*, 14:691–707, 1986.

[20] Feigin, P. D. and A. Cohen, On a model for concordance between judges. *J. Royal Statistical Soc. Series B*, 40:203–213, 1978.

[21] Fligner, M. A. and J. S. Verducci, Distance based ranking models. *J. Royal Statistical Soc. Series B*, 48:359–369, 1986.

[22] Gordon, A. D., Another measure of the agreement between rankings. *Biometrika*, 66(2):327–332, 1979.

[23] Gordon, A. D., A measure of the agreement between rankings. *Biometrika*, 66(1):7–15, 1979.

[24] Graham, R. L., D. E. Knuth and O. Patashnik, *Concrete Mathematics*. Addison-Wesley Publishing Co., Reading, MA, 1989.

[25] Guibas, L. J., E. M. McCreight and M. F. Plass, A new representation of linear lists. In *The Proceedings of the 9th ACM Annual Symposium on Theory of Computing*, pages 49–60, 1977.

[26] Hadjicostas, P. and K. B. Lakshmanan, Bubblesort with erroneous comparisons. www.acs.brockport.edu/~klakshma/bio/research.html, 2003.

[27] Hammersley, J. M., A few seedlings of research. *Proc. Sixth Berckley Symp. Math. Statis. Probability*, 1:345–394, 1972. Univ. of California Press.

[28] Hertel, S., Smoothsort's behavior on presorted sequences. *Information Processing Letters*, 16:165–170, 1983.

[29] Islam, T. and K. B. Lakshman, On the error-sensitivity of sort algorithms. In S. G. Akl, F. Fiala, and W. W. Koezkodaj, editors, *Proceedings of the International Conference On Computing and Information*, pages 81–85, Toronto, May 1990. Canadian Schoolar's Press. Advances in Computing and Information.

[30] Katajainen, J.,C. Levcopoulos and O. Petersson, Local insertion sort revisited. In *Proc. Optimal Algorithms*, pages 239–253. Springer-Verlag Lecture Notes in Computer Science 401, 1989.

[31] Kendall, M. G., *Rank Correlation Methods*. Griffin, London, 4th edition, 1970.

[32] Kingman, J. F. C., Subadditive ergodic theory. *Annals of Probability*, 1:883–899, 1973.

[33] Knuth, D. E., *The Art of Computer Programming, Vol.2: Seminumerical Algorithms*. Addison-Wesley Publishing Co., Reading, MA, 1973.

[34] Knuth, D. E., *The Art of Computer Programming, Vol.3: Sorting and Searching*. Addison-Wesley Publishing Co., Reading, MA, 1973.

[35] Lenstra, J. K., Clustering a data array and the traveling-salesman problem. *Operations Research*, 22:413–414, 1974.

[36] Levcopoulos, C., and O. Petersson, Heapsort — adapted for presorted files. In F. Dehne, J.R. Sack, and N. Santoro, editors, *Proceedings of the Workshop on Algorithms and Data Structures*, pages 499–509. Springer-Verlag Lecture Notes in Computer Science 382, 1989.

[37] Levcopoulos, C. and O. Petersson, Sorting shuffled monotone sequences. Technical report, Department of Computer Science, Lund University, Box 118, S-2100 Lund, Sweden, 1989.

[38] Levcopoulos, C. and O. Petersson, Splitsort—an adaptive sorting algorithm. In B. Rovan, editor, *Mathematical Foundations of Computer Science*, pages 416–422. Springer-Verlag Lecture Notes in Computer Science 452, 1990.

[39] Levcopoulos, C. and O. Petersson, Adaptive Heapsort. *Journal of Algorithms*, 14:395–413, 1993.

[40] B.F. Logan and L.A. Shepp. A variational problem for random Young tableaux. *Advances in Mathematics*, 26:206–22, 1977.

[41] Mallows, C. L., Non-null ranking models. *Biometrika*, 44:114–130, 1957.

[42] Mannila, H., *Instance Complexity for Sorting and NP-Complete Problems.* PhD thesis, University of Helsinki, Department of Computer Science, 1985.

[43] Mannila, H., Measures of presortedness and optimal sorting algorithms. *IEEE Transactions on Computers*, C-34:318–325, 1985.

[44] McCormick, Jr., H. T., P. J. Schweitzer and T. W. White, Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20:993–1009, 1972.

[45] McKay, J., The largest degrees of irreducible characters of the symmetric group. *Mathematics of Computation*, 30(135):624–635, 1976.

[46] Mehlhorn, K., Sorting presorted files. *Proceedings of the 4th GI Conference on Theory of Computer Science*, Springer-Verlag Lecture Notes in Computer Science 67:199–212, 1979.

[47] Mehlhorn, K., *Data Structures and Algorithms, Vol 1: Sorting and Searching.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin/Heidelberg, 1984.

[48] Messmer, B. T. and H. Bunke, A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, pages 1979–1998, 1999.

[49] Moffat, A. Eddy and O. Pettersson, Splaysort. fast, versatile, practical. *Software – Practice and Experience*, 26(7):781–797, July 1996.

[50] Oommen, B. J., and D. T. Ng, On generating random permutations with arbitrary distributions. *The Computer Journal*, 33(4):368–374, 1990.

[51] Özsu, M. T., and P. Valduriez, *Principles of Distributed Database Systems.* Prentice Hall, 2nd edition, 1999.

[52] Ross, S. M., *Introduction to Probability Models.* Academic Press, Inc., Orlando, Florida, 3rd edition, 1985.

[53] Roura, S. Improving Mergesort for linked lists. In J. Nesetril, editor, *Algorithms - ESA '99, 7th Annual European Symposium*, volume 1643, pages 267–276, Prague, Czech Republic, July 16-18 1999. Lecture Notes in Computer Science.

[54] Sedgewick, R., *Quicksort.* Garland Publishing Inc., New York and London, 1980.

[55] Skiena, S. S., Encroaching lists as a measure of presortedness. *BIT*, 28:755–784, 1988.

[56] Sloane, N. J. A., Encrypting by random rotations. In T. Beth, editor, *Proceedings of Cryptography, Burg Feuerstein 82*, pages 71–128. Springer-Verlag Lecture Notes in Computer Science 149, 1983.

[57] Spearman, C., The proof and measurement of association between two things. *American Journal of Psychology*, 15(1):72–101, 1904.

[58] Stanley, R. P., On planar partitions: Part II. *Studies in Applied Math.*, 50:259–279, 1971.

[59] Tsai, W. H. and K.-S. Fu, Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Transactions on Systems, Man and Cybernetics*, 9(12):757–768, December 1979.

[60] Vershik, A. M. and S. V. Kerov, Asymptotics of the Placherel measure of the symmetric group and the limiting form of Young tables. *Soviet Math. Dokl.*, 18:527–531, 1977.

[61] Wang, Y.-K., K.-C. Fan and J.-T. Horng, Genetic-based search for error-correcting graph isomorphism. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 27(4):588–597, August 1997.

[62] Xiao, J., Y. Zhang, X. Jia and T. Li, Measuring similarity of interests for clustering web-users. In *Australian Computer Science Communications. Database Technologies. Proceedings of the 12th Australasian Database Conference ADC 2001*, volume 23-2, pages 107–114. IEEE Computer Society, 2001.

7 Appendix

We present here laborious proofs for completeness so referees can review them. We hope that placing them here facilitates the flow of the main ideas. For Theorem 3.2.

Proof. It is not difficult to verify axioms 1, 2 and 4. To verify axiom 3, we show that if Y is a subsequence of X obtained by throwing away one element, then $\|Y\|_p \leq \|X\|_p$, for all $p \geq 1$. The general case follows by induction.

Let $p \geq 1, p \in \Re$. Let $X = \langle x_1, x_2, \ldots, x_n \rangle$, and let Y be obtained from X by deleting x_i, thus $Y = \langle x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots x_n \rangle$. Assume $Perm[X](i) > i$ (the case $Perm[X](i) < i$ is symmetric, and if $Perm[X](i) = i$ then $\|Y\|_p \leq \|X\|_p$ trivially).

Let $m = |Perm[X](i) - i|$ and consider $(\|X\|_p)^p = \sum_{j=1}^{n} |Perm[X](j) - j|^p$. Since

$$y_j = \begin{cases} x_j & \text{if } j \leq i-1 \\ x_{j+1} & \text{if } j \geq i \end{cases}$$

and

$$Perm[Y](j) = \begin{cases} Perm[X](j) & \text{if } j < i \text{ and } Perm[X](j) < i \\ Perm[X](j+1) & \text{if } j \geq i \text{ and } Perm[X](j) < i \\ Perm[X](j) - 1 & \text{if } j < i \text{ and } Perm[X](j) \geq i \\ Perm[X](j+1) - 1 & \text{if } j \geq i \text{ and } Perm[X](j) \geq i \end{cases}$$

the terms $|Perm[X](j) - j|$ with $j \neq i$ in $(\|X\|_p)^p$ are in a one to one correspondence with the terms in $(\|Y\|_p)^p$. That is, $(\|Y\|_p)^p = \sum_{j=1}^{n-1} |Perm[Y](j) - j|^p$, and each $Perm[Y](j)$ can be rewritten by $Perm[X](j)$ with $j \neq i$. A careful case analysis shows that the corresponding term to $|Perm[Y](j) - j|$ is always larger unless $i \leq j < Perm[Y](j) < Perm[X](i)$ and in this case, it increases by one. More precisely, using the correspondence between terms and Iverson's notation [24, Chapter 2] (for a predicate P, $[P] = 1$ if P is true and $[P] = 0$ if P is false), we write

$$A_X = \sum_{j=1}^{i-1} |Perm[X](j) - j|^p [Perm[X](j) < Perm[X](i)],$$

$$B_X = \sum_{j=1}^{i-1} |Perm[X](j) - j|^p [Perm[X](j) > Perm[X](i)],$$

$$\begin{aligned}
C_{X_1} &= \sum_{j=i+1}^{n} |Perm[X](j) - j|^p [Perm[X](j) < Perm[X](i)] \\
&\quad [j \leq Perm[X](j)], \\
C_{X_2} &= \sum_{j=i+1}^{n} |Perm[X](j) - j|^p [Perm[X](j) < Perm[X](i)] \\
&\quad [j > Perm[X](j)], \\
D_X &= \sum_{j=i+1}^{n} |Perm[X](j) - j|^p [Perm[X](j) > Perm[X](i)], \\
A_Y &= \sum_{j=1}^{i-1} |Perm[Y](j) - j|^p [Perm[X](j) < Perm[X](i)], \\
B_Y &= \sum_{j=1}^{i-1} |Perm[Y](j) - j|^p [Perm[X](j) < Perm[X](i)], \\
C_{Y_1} &= \sum_{j=i}^{n-1} |Perm[Y](j) - j|^p [Perm[X](j+1) < Perm[X](i)] \\
&\quad [j+1 \leq Perm[X](j+1)], \\
C_{Y_2} &= \sum_{j=i}^{n-1} |Perm[Y](j) - j|^p [Perm[X](j+1) < Perm[X](i)] \\
&\quad [j+1 > Perm[X](j+1)], \\
D_Y &= \sum_{j=i}^{n-1} |Perm[Y](j) - j|^p [Perm[X](j+1) > Perm[X](i)].
\end{aligned}$$

Therefore,

$$(\|X\|_p)^p = A_X + B_X + C_{X_1} + C_{X_2} + D_X + m^p,$$

$$(\|Y\|_p)^p = A_Y + B_Y + C_{Y_1} + C_{Y_2} + D_Y$$

and

$$A_Y = A_X, \quad B_Y \leq B_X, \quad C_{Y_2} \leq C_{X_2} \text{ and } \quad D_Y = D_X.$$

Thus, in order to prove that $(\|Y\|_p)^p \leq (\|X\|_p)^p$ it is sufficient to prove that $C_{Y_1} - C_{X_1} \leq m^p$. Let $J = \{j \in [1,n] \mid i < j \leq Perm[X](j) < Perm[X](i)\}$, and $R = \{Perm[X](j) - j \quad | \; j \in J\}$. The cardinality of R is less than $Perm[X](i) - i$; that is, $\|R\| \leq m - 1$ and if $r \in R$ then, $|r| \leq m - 2$. If we denote $R = \{r_1, r_2, \ldots, r_t\}$ then, $C_{X_1} = \sum_{k=1}^{t} r_k^p$ and $C_{Y_1} = \sum_{k=1}^{t} (r_k + 1)^p$. For $w \geq 0$, $f(w) = w^p$ is convex and monotonically increasing, thus $0 \leq u < v$

implies $f(v) - f(u) \leq f'(v)(v-u)$. Therefore,

$$C_{Y_1} - C_{X_1} = \sum_{k=1}^{t}[(r_k+1)^p - r_k^p] \leq \sum_{k=1}^{t} p(r_k+1)^{p-1}.$$

Case $p \geq 2$: Under the conditions of this problem, it can be shown (see below) that

$$\max\left(\sum_{k=1}^{t}(r_k+1)^{p-1}\right) \leq \frac{m^p}{p}, \tag{2}$$

thus

$$C_{Y_1} - C_{X_1} \leq p\frac{m^p}{p} = m^p,$$

as required.
Case $1 \leq p \leq 2$: Under the conditions of this problem, it can be shown (see below) that

$$\max\left(\sum_{k=1}^{t}(r_k+1)^{p-1}\right) \leq \max_{t\in\{1,2,\ldots,m-1\}} t(m-t)^{p-1}. \tag{3}$$

Let $g(t) = t(m-t)^{p-1}$, $t \in [1, m-1]$. Setting the derivative of $g(t)$ equal to zero and solving for t, shows that $g(t)$ is maximized when $t = m/p$. This implies that

$$\begin{aligned} C_{Y_1} - C_{X_1} &\leq p \max_{t\in\{1,2,\ldots,m-1\}} t(m-t)^{p-1} \\ &\leq p(m/p)(m-m/p)^{p-1} = m^p(\frac{p-1}{p})^{p-1} \leq m^p \end{aligned}$$

as required. □

To prove (2) and (3) we translate the problem of bounding $\max(\sum_{k=1}^{t}(r_k+1)^{p-1})$ into the problem of bounding the weight of the matchings in a weighted bipartite graph. We introduce the corresponding graph in Definition 7.1 and the required results in Lemmas 7.2 and 7.3.

Definition 7.1 Let $q \geq 0, q \in \Re$, $m \geq 2$ and let $G_q(V,E)$ be the weighted bipartite graph given by: $V = V_1 \cup V_2$, $|V_1| = |V_2| = m-1$, $V_1 = \{u_1, u_2, \ldots, u_{m-1}\}$, $V_2 = \{v_1, v_2, \ldots, v_{m-1}\}$ and $(u_i, v_j) \in E$ if and only if $i \leq j$ and, in that case, the weight of the edge is $c[(u_i, v_j)] = (j-i+1)^q$.

Now, let u_k correspond to $i+k$ and v_k correspond to $i+k$, for $k = 1, 2, \ldots, m-1$. The set R defines a matching in G_{p-1} by: if $Perm[X](j) - j \in R$, then include the edge $(u_{j-i}, v_{Perm[X](j)-j})$ in the matching. Now, $\sum_{k=1}^{t}(r_k+1)^{p-1}$ is the weight of the matching.

To obtain (2) apply Lemma 7.2 with $q = p - 1$. For (3) use Lemma 7.3.

Lemma 7.2 *Let $q \geq 1$, $q \in \Re$, and G_q as in Definition 7.1.*

(i) *A matching of maximum weight is given by:*

$$M = \{(u_1, v_{m-1}), (u_2, v_{m-2}), \dots, (u_{\lfloor m/2 \rfloor}, v_{m-\lfloor m/2 \rfloor})\}.$$

(ii) *The cost of the maximum weight matching is no greater than $\frac{m^{q+1}}{q+1}$.*

Proof. We prove the lemma by induction on m.
Basis: For $m = 2$ the graph has only one edge, this edge is a matching of maximum weight. For $m = 3$ the graph has four non-empty matchings:

M	weight
$\{(u_1, v_1)\}$	1^q
$\{(u_2, v_2)\}$	1^q
$\{(u_1, v_1), (u_2, v_2)\}$	$1^q + 1^q$
$\{(u_1, v_2)\}$	2^q

Clearly the claim holds.
Induction Step: Assume the claim is true for k, $2 \leq k < m$. We will show that any matching M of maximum weight can be transformed, without reducing the weight, into a matching M' containing the edge (u_1, v_{m-1}). Thus M' is a matching of maximum weight. To obtain a maximum weight matching in the form claimed by the lemma, we observe that $M' - \{(u_1, v_{m-1})\}$ is a matching of the subgraph G'_q obtained from G_q by removing the vertices $u_1, v_1, u_{m-1}, v_{m-1}$. But (u_1, v_{m-1}) is the largest edge in G_q thus applying the induction hypothesis to G'_q proves the result.

It only remains to prove that (u_1, v_{m-1}) belongs to a matching of maximum weight. Let M be a matching of maximum weight that does not include the edge (u_1, v_{m-1}). Let u_s be such that $(u_s, v_{m-1}) \in M$, $1 < s \leq m - 1$. (If no edge in M is adjacent to v_{m-1}, then $M \cup \{(u_{m-1}, v_{m-1})\}$ will be a matching of larger weight since in G_q, u_{m-1} is only adjacent to v_{m-1}.) Let v_r be such that $(u_1, v_r) \in M$, $1 \leq r < m - 1$. (Again, if no edge in M is adjacent to u_1, then $M \cup \{(u_1, v_1)\}$ will be a matching of larger weight since in G_q, v_1 is only adjacent to u_1.)
Case 1: $r < s$. Let $M' = M \cup \{(u_1, v_{m-1})\} - \{(u_s, v_{m-1}), (u_1, v_r)\}$. Then, clearly M' is a matching and

$$\begin{aligned} W(M') &= W(M) + (m-1)^q - (m-1-s+1)^q - (r-1+1)^q \\ &= W(M) + (m-1)^q - ((m-s)^q + r^q). \end{aligned}$$

Since for $q \geq 1$, $a, b \geq 0$ we have $a^q + b^q \leq (a+b)^q$, we obtain

$$(m-s)^q + r^q \leq (m-s)^q + (s-1)^q \leq (m-1)^q$$

and $W(M') \geq W(M)$ as claimed.
Case 2: $r \geq s$. Let $M' = M \cup \{(u_1, v_{m-1}), (u_s, v_r)\} - \{(u_s, v_{m-1}), (u_1, v_r)\}$. Then, clearly M' is a matching and

$$W(M') = W(M) + (m-1)^q + (r-s+1)^q - ((m-s)^q + r^q).$$

Since for $q \geq 1$, $0 \leq c \leq a, b$ we have $a^q + b^q \leq c^q + [a + (b-c)]^q$, we obtain

$$(m-s)^q + r^q \leq (r-s+1)^q + [(m-s) + (r-(r-s+1))]^q = (r-s+1)^q + (m-1)^q$$

and $W(M') \geq W(M)$ as claimed.

Now, let M be a maximum weight matching.

$$\begin{aligned} W(M) &= \sum_{i=1}^{\lfloor m/2 \rfloor} (m-i-i+1)^q \\ &= \sum_{i=1}^{\lfloor m/2 \rfloor} (m-2i+1)^q \leq \sum_{i=0}^{m-1} i^q \leq \int_0^{m-1} (x+1)^q dx = \frac{m^{q+1}}{q+1} \end{aligned}$$

□

Lemma 7.3 *Let $0 \leq q < 1, q \in \Re$, and G_q be as in Definition 7.1. Then,*

(i) *there is $r \geq 1$ such that a matching of maximum weight is given by:*

$$M = \{(u_1, v_{m-r}), (u_2, v_{m+1-r}), \ldots, (u_r, v_{m-1})\}$$

and,

(ii) *there is $r \geq 1$ such that $r \leq m-1$ and the cost of the maximum weight matching is no greater than than $r(m-r)^q$.*

We leave the proof of this last lemma to the reader.

For Theorem 3.3.

Proof. Let $M(X) = \sum_{i=1}^{r} l_i M_i(X)$, where l_i are non-negative constants l_i, and M_i are measures of disorder. By definition, M depends only on the relative order of the elements in X and since, for all i, $M_i(X)$ is minimized when X is sorted, and l_i are nonnegative, $M(X)$ is minimized when X is sorted. Therefore, M is a *mod.*

If Y is a subsequence of X, then $M_i(Y) \leq M_i(X)$, for all i; this gives $M(Y) = \sum_{i=1}^{r} l_i M_i(Y) \leq \sum_{i=1}^{r} l_i M_i(X) = M(X)$.

Now, because for the M_i's satisfy condition 3 $Y \leq X$, implies that $M_i(YX) \leq M_i(Y)+M_i(X)$, for all i. Therefore, $M(YX) = \sum_{i=1}^{r} l_i M_i(YX) \leq \sum_{i=1}^{r} l_i[M_i(Y)+M_i(X)]$. But then, $M(YX) = \sum_{i=1}^{r} l_i M_i(Y)+\sum_{i=1}^{r} l_i M_i(X) = M(Y)+M(X)$.

Finally,

$$\begin{aligned} M(\langle x\rangle X) &= \sum_{i=1}^{r} l_i M_i(\langle x\rangle X) \\ &\leq \sum_{i=1}^{r} l_i[c_i|X| + M_i(X)] \\ &= \left(\sum_{i=1}^{r} l_i c_i\right) |X| + M(X). \end{aligned}$$

□

To prove *EncR* is a measure of presortedness we first prove the following lemma.

Lemma 7.4 *If Y is a subsequence of X, then $Enc(Y) \leq Enc(X)$.*

Proof. Assume Y is a subsequence of $X = \langle x_1, \ldots, x_n\rangle$ obtained by deleting x_i from X. The general case follows by induction. Let $D(x_i)$ denote the dequeue containing x_i in the encroaching set for X. For $x_1, \ldots, x_{i-1}$, the encroaching sets of Y and X are the same, and elements in $x_{i+1}, \ldots, x_n$ that are blocked by x_i end up in dequeues no older than those obtained when building the encroaching set for Y. Thus, no more dequeues are required for Y than for X. □

Proof. The definition of $EncR(X)$ depends on only the relative order of elements in X and, clearly, if X is sorted, then $EncR(X) = 0$; thus, we verify axioms 3, 4 and 5 directly.

We verify axiom 3 in Definition 1.5; that is, if Y is a subsequence of X, then $EncR(Y) \leq EncR(X)$. Suppose Y is a subsequence of X. If Y is sorted, then $EncR(Y) = 0 \leq EncR(X)$. If Y is not sorted, then P_Y is a subsequence of P_X and by Lemma 7.4

$$Enc(Reverse(P_Y)) \leq Enc(Reverse(P_X)).$$

We now prove axiom 4, namely, if $X \leq Y$, then $EncR(XY) \leq EncR(X)+EncR(Y)$. Let $X \leq Y$. If Y is sorted, then $P_X = P_{XY}$ and

$$EncR(XY) = Enc(Reverse(P_X)) = EncR(X) \leq EncR(X) + EncR(Y).$$

Assume Y is not sorted. The encroaching set for $Enc(XY)$ is built by constructing the encroaching set of $Reverse(P_Y)$ and then adding to it the elements in $Reverse(X)$. Therefore, $X \leq Y$ implies $S_X \leq P_Y$, and the elements of the sorted sequence S_X end up in the first dequeue. Thus, there is a subsequence B of P_X such that

$$EncR(XY) = Enc(Reverse(B)) + Enc(Reverse(P_Y)).$$

By Lemma 7.4, $Enc(Reverse(B)) \leq Enc(Reverse(P_X))$; thus,

$$\begin{aligned} EncR(XY) &\leq Enc(Reverse(P_X)) + EncR(Y) \\ &= EncR(X) + EncR(Y). \end{aligned}$$

We now check axiom 5, namely, $EncR(\langle x\rangle X) \leq |X| + EncR(X)$. We analyze two cases.

Case $X = \langle\rangle$. This implies $EncR(\langle x\rangle X) = 0 = |X| + EncR(X)$, as required.
Case $X \neq \langle\rangle$. There are three subcases.
Subcase $|X| = 1$. This implies $EncR(\langle x\rangle X) \leq 1 \leq |X| + EncR(X)$;
Subcase X is sorted and $|X| > 1$. This implies $Enc(\langle x\rangle X) \leq 2 \leq |X| + EncR(X)$;
Subcase X is not sorted. This implies $P_{\langle x\rangle X} = \langle x\rangle P_X$. Thus,

$$\begin{aligned} Enc(Reverse(P_{\langle x\rangle X})) &\leq Enc(Reverse(P_X)) + 1 \\ &\leq EncR(X) + |X|. \end{aligned}$$

This completes the proof. □