INFORMATION AND CONTROL 61, 114–131 (1984)

An Average Case Analysis of Floyd's Algorithm to Construct Heaps

ERNST E. DOBERKAT

Department of Mathematics and Computer Science, Clarkson College of Technology, Potsdam, New York 13676

The expected number of interchanges and comparisons in Floyd's well-known algorithm to construct heaps and derive the probability generating functions for these quantities are considered. From these functions the corresponding expected values are computed.

1. INTRODUCTION

We are going to investigate the expected numbers of interchanges and comparisons for Floyd's algorithm for heap construction (Knuth, 1973b, 5.2.3; Floyd, 1964). This will be done by computing the corresponding probability generating functions, from which the looked for values will be obtained, and from which in principle information on all the higher moments could be derived.

Suppose we are given an array $x[1 \cdots N]$ such that $x[i]$ is the label of node i, when $\{1,...,N\}$ is represented in the canonical way as a binary tree (i.e., 1 is the root of this tree, and $2i$ and $2i+1$ are, respectively, the left and the right sons of node i). This array is called a heap if each node has a greater label than its father, or, more formally, iff

$$x[i] > x[\lfloor i/2 \rfloor]$$

holds for every $i \in \{2,...,N\}$. Floyd's algorithm for heap construction then reads as given below (note that we here take the elegant recursive version as given in (Aho, Hopcroft, and Ullman, 1974); Floyd's original formulation is iterative rather than recursive):

Input: the array x
Output: x, organized as a heap
Method: 0. **procedure** heapify(k);
if k is no leaf and a son of k has a smaller label than k has
then let j be the son with the smallest label;
interchange $x[k]$ with $x[j]$;
heapify(j)

0019-9958/84 $3.00

```
    endif
    end heapify;
1.  for k := ⌊N/2⌋ downto 1 do heapify(k).
```

Thus if k is a node with the property that both its left and its right subtree have the heap property already (i.e., every node has a smaller label than its offsprings), a call to heapify(k) will check whether this condition is met for k. If this is the case, the next node will be processed, if not, the label for k will be interchanged with that for the son having the smallest label, and this consideration is repeated for that node. In this way every node starting from the rightmost deepest nonleaf and proceeding from right to left to the root is made the root of a heap.

This algorithm has the very convenient property of preserving randomness; this means that uniformly distributed input data will produce uniformly distributed heaps. Knuth derives this fact for uniformly distributed permutations of $\{1,\ldots,N\}$ and uses this to derive a number of probabilistic characteristics for the algorithm, e.g., the expected number of times the left son has a larger label than the right son or the expectation for the total number of keys promoted during a call to heapify (Knuth, 1973b, 153–157). These investigations are based on the geometric structure of the underlying tree. We make use of these results concerning stability of the distribution in a considerably more general continuous probabilistic model and carry the analysis a bit further by considering expectations for interchanges and comparisons.

The paper is organized as follows: in Section 2 heaps are introduced more formally and the probabilistic assumptions are made explicit. Section 3 derives the probability generating functions for interchanges and for comparisons, from which the wanted expectations are computed.

2. Heaps

Represent $\{1,\ldots,N\}$ as a tree as mentioned in the Introduction. We will make heavy use of this tree representation in the sequel and quote some notations and results from (Knuth, 1973b, 153–157). The path leading from N to 1 is called the special path; if $N = (1b_{n-1}\cdots b_0)_2$ in binary, then

$$\{(1b_{n-1}\cdots b_j)_2; 0 \leqslant j \leqslant n\}$$

constitutes this special path; note that $j = n$ gives the root 1 of the tree. The nodes of the special path are called special nodes. If k is no special node, then the subtree rooted at k is complete. Let $\gamma(k)$ be the size of the subtree rooted at k, then

$$\gamma((1b_{n-1}\cdots b_j)_2) = (1b_{j-1}\cdots b_0)_2 .$$

Call a node k on level j a left (right) node, if $2^j \leqslant k < (1b_{n-1} \cdots b_{n-j})_2$ ($(1b_{n-1} \cdots b_{n-j})_2 < k \leqslant 2^{j+1} - 1$) holds; thus a left (right) node lies to the left (right) of the special path. It is clear that there are

$$(1b_{n-1} \cdots b_{n-j})_2 - 2^j$$

left nodes and

$$2^{j+1} - 1 - (1b_{n-1} \cdots b_{n-j})_2$$

right nodes on level j. The subtree rooted at a left (right) node on this level has $2^{n+1-j} - 1$ ($2^{n-j} - 1$) nodes. We will use these numbers when calculating the generating functions below.

It is assumed that the inputs to the algorithm are taken from some set in Euclidean space and are continuously distributed. To be more specific, we assume that the input space A is a symmetric set in the following sense:

(a) $A \subset \mathbb{R}^N$ is a Borel set,

(b) if $x \in A$ then $x_i \neq x_j$ for $i \neq j$,

(c) if $x \in A$ then $(x_{p(1)}, \ldots, x_{p(N)}) \in A$ for every $p \in \mathscr{S}_N := \{p;\ p$ is a permutation of $\{1, \ldots, N\}\}$.

Condition (a) is a technical one allowing us to define a probability measure on A, condition (c) will prevent that the outputs of the algorithm will escape from A (thus preventing technical difficulties), and condition (b) which will hold almost everywhere for a continuous probability distribution on A, is included for convenience. The distribution μ of the elements of A is assumed to have the density f, thus

$$\mu(B) = \int_B f(x)\, dx$$

holds for every Borel set $B \subset A$, where f is symmetric in the sense that

$$\forall x \in A \forall p \in \mathscr{S}_N : f(x_1, \ldots, x_N) = f(x_{p(1)}, \ldots, x_{p(N)})$$

holds. Thus f will have the same value for a vector x and for all vectors that are obtained from x by permuting components. The pair (A, μ) is called a symmetric model in (Doberkat, 1983) and discussed there extensively; examples for these models may be found there, too. For the remainder of this paper a symmetric model (A, μ) is fixed.

In order to have a look at some distributional properties of the algorithm, k-heaps are remembered from (Knuth, 1973b): a vector $x \in A$ is called a k-heap iff the subtree rooted at k has the heap property; A_k is the set of all k-

heaps. Given an input x, denote the resulting k-heap by $x^{(k)}$; using the procedure heapify, $x^{(k)}$ is obtained from x by executing

$$\textbf{for } j := \lfloor N/2 \rfloor \textbf{ downto } k \textbf{ do } \text{heapify}(j).$$

Now suppose that $y \in A_{k+1}$ is a $(k+1)$-heap, then a call to heapify(k) will make it a k-heap, and y_k as the original label of node k will label another node j upon return from this procedure call, where j is in the subtree rooted at k. Let $A_{k,j}$ denote the set of all those $(k+1)$-heaps with this property, then the following is not difficult to establish:

LEMMA 2.1. *Let $T_k : A_{k+1} \to A_k$ be the map corresponding to heapify(k), then $T_k : A_{k,j} \to A_k$ is a bijection for every j in the subtree rooted at k.*

This implies that every k-heap has exactly $\gamma(k)$ inverse images under T_k (or under heapify(k)). Now let $\mu^{(k)}$ be the distribution of all k-heaps, then Lemma 2.1 will be useful to characterize $\mu^{(k)}$. More formally, $\mu^{(k)}$ may be defined inductively by

$$\mu^{(\lfloor N/2 \rfloor + 1)} := \mu,$$
$$\mu^{(k)} := T_k(\mu^{(k+1)}),$$

thus $\mu^{(k)}$ is the image measure of $\mu^{(k+1)}$ under T_k.

PROPOSITION 2.2.

$$\mu^{(k)} = \prod_{i=k}^{\lfloor N/2 \rfloor} \gamma(j) \cdot \mu.$$

Proof. Define the order type $\alpha(x)$ of a vector $x \in A$ as the unique permutation p such that $x_{p(1)} > \cdots > x_{p(N)}$ holds. In Section 4 of (Doberkat, 1983) it is shown that a symmetric model yields the same distributioon of order types for any algorithm that manipulates its inputs based only on their order type, as the discrete model of randomness does. Thus the proposition follows from Theorem 5.2.3. H in (Knuth, 1973b). ∎

Hence the algorithm in question preserves the originally given distribution up to a weight factor. Note that Proposition 2.2 may be derived without recurring to uniformly distributed permutations using some machinery from measure theory, see (Doberkat, 1980).

The following consequence will be important in the sequel

COROLLARY 2.3. $\mu^{(k+1)}(A_{k,j}) = \gamma(k)^{-1}$ *for every j in the subtree rooted at k.*

Proof. Abbreviating $\prod_{j=k}^{\lfloor N/2 \rfloor} \gamma(j)$ by C_k, we see that

$$\begin{aligned}
\mu^{(k+1)}(A_{k,j}) &= C_{k+1}\mu(A_{k,j}) \\
&= C_{k+1} \cdot \int_{A_{k,j}} f(x)\,dx \\
&\overset{(*)}{=} C_{k+1} \cdot \int_{A_k} f(x)\,dx \\
&= \gamma(k)^{-1} \cdot C_k \cdot \mu(A_k) \\
&= \gamma(k)^{-1}.
\end{aligned}$$

The equality (*) holds by the change of variables formula (Rudin, 1974, p. 186), the symmetry of f, and because of Lemma 2.1. The Jacobian of T_k is identical to 1 since T_k only permutes coordinates, thus is an orthogonal transformation. ∎

With these tools at hand, we are ready to compute some probability generating functions.

3. Generating Functions for Interchanges and Comparisons

Suppose we execute heapify(k) on $y \in A_{k,j}$, then the distance $d(k,j)$ between k and j will give the number of interchanges. Consequently, Corollary 2.3 implies that

$$\mathscr{E}_k(z) := \gamma(k)^{-1} \cdot \sum_{t=0}^{\infty} |\{j; j \text{ is in the subtree rooted at } k, d(k,j) = t\}| \cdot z^t.$$

Denote by $f(k,j)$ the number of comparisons for $y \in A_{k,j}$, then the corresponding generating function for comparisons may be written as

$$\mathscr{F}_k(z) := \gamma(k)^{-1} \cdot \sum_{t=0}^{\infty} |\{j; j \text{ is in the subtree rooted at } k, f(k,j) = t\}| \cdot z^t.$$

The function f may be determined as follows: if $k = j = 1$ (and $N \geqslant 3$), put $f(1, 1) := 2$, otherwise put

$$f(k, j) := 2(d(k, j) - 1) + s(\lfloor j/2 \rfloor) + s(j),$$

where $s(j)$ is the number of sons of node j. This takes into account that $\lfloor N/2 \rfloor$ has only one son in case N is even. Note that

$$f(k, j) = 2d(k, j) + s(j)$$

holds provided N is odd, or k is not on the special path, if N is even.

The corresponding generating functions for the interchanges and comparisons done by the algorithm as a whole are a bit trickier to obtain, since we have to deal with possibly variing depths of recursion, as the k-loop proceeds.

THEOREM 3.1. *If $\mathscr{E}$ and $\mathscr{V}$ are the probability generating functions for interchanges and comparisons for Floyd's algorithm, then*

$$\mathscr{E}(z) = \prod_{k=1}^{N} \mathscr{E}_k(z)$$

and

$$\mathscr{V}(z) = \prod_{k=1}^{N} \mathscr{V}_k(z)$$

hold.

Proof. (1) Let j_k be an arbitrary node in the subtree rooted at k, $1 \leqslant k \leqslant \lfloor N/2 \rfloor$ and define $P(j_1,..., j_{\lfloor N/2 \rfloor}) := \{x \in A; x^{(k+1)} \in A_{k,j_k} \text{ for } 1 \leqslant k \leqslant \lfloor N/2 \rfloor\}$, then evidently

$$\mathscr{P} := \{P(j_1,..., j_{\lfloor N/2 \rfloor}); j_k \text{ is in the subtree rooted at } k \text{ for } 1 \leqslant k \leqslant \lfloor N/2 \rfloor\}$$

is a partition of A consisting of $h := \prod_{j=1}^{\lfloor N/2 \rfloor} \gamma(j)$ members. Note that $x \in P(j_1,..., j_{\lfloor N/2 \rfloor})$ implies that exactly

$$\sum_{i=1}^{\lfloor N/2 \rfloor} d(i, j_i)$$

interchanges will be done. Because of Lemma 2.1, it is not difficult to establish that there exists for every $P \in \mathscr{P}$ a bijection $P \to P(1, 2,..., \lfloor N/2 \rfloor)$ that permutes components only. Thus by the change of variables formula we may deduce that

$$\mu(P) = \mu(P(1, 2,..., \lfloor N/2 \rfloor))$$

holds for every partition element. Consequently, $\mu(P) = h^{-1}$ holds for all $P \in \mathscr{P}$.

(2) This enables us to compute the generating function for

interchanges. For this, abbreviate by T the number of interchanges. Then we have for $t \geqslant 0$,

$$\begin{aligned}\mu(T=t) &= \sum \{\mu(P(j_1,\ldots,j_{\lfloor N/2\rfloor})); \sum_{i=1}^{\lfloor N/2\rfloor} d(i,j_i)=t\} \\ &= h^{-1}\cdot|\{(j_1,\ldots,j_{\lfloor N/2\rfloor}); \sum_{i=1}^{\lfloor N/2\rfloor} d(i,j_i)=t\}| \\ &= \text{coefficient of } z^t \quad \text{in} \quad \prod_{k=1}^{N} \mathscr{E}_k(z).\end{aligned}$$

(Note that $\mathscr{E}_k(z)=1$ if $k > \lfloor N/2\rfloor$). Replacing $d(i,j)$ by $f(i,j)$, the result for comparisons follows in a similar way. ∎

Consequently, we see that there will be

$$\sum_{k=1}^{N} \mathscr{E}_k'(1)$$

interchanges and

$$\sum_{k=1}^{N} \mathscr{V}_k'(1)$$

comparisons on the average with a variance of

$$\sum_{k=1}^{n} [\mathscr{E}_k''(1)+\mathscr{E}_k'(1)-(\mathscr{E}_k'(1))^2]$$

and

$$\sum_{k=1}^{n} [\mathscr{V}_k''(1)+\mathscr{V}_k'(1)-(\mathscr{V}_k'(1))^2],$$

respectively (see Knuth 1973a, p. 98).

Let us compute $\mathscr{E}$ and $\mathscr{V}$ in the special case that $N=2^{n+1}-1$, thus every node has either two or no offsprings. Suppose k is at level j, so $k=2^j+i$ for some i, $0 \leqslant i \leqslant 2^j-1$. Given t with $0 \leqslant t \leqslant n-j$, there are 2^t nodes in the subtree rooted at k the distance of which to k equals t. Since $\gamma(k)=2^{n+1-j}-1$, we see that

$$\mathscr{E}_{2^j+i}(z) = \frac{1}{2^{n+1-j}-1}\sum_{t=0}^{n-j} 2^t z^t = \frac{1}{2^{n+1-j}-1}\cdot\frac{(2z)^{n+1-j}-1}{2z-1}$$

holds. Since the 2^j subtrees at this level all have the same shape,

$$\mathscr{E}(z) = \prod_{j=0}^{n-1} \left[\frac{1}{2^{n+1-j}-1} \cdot \frac{(2z)^{n+1-j}-1}{2z-1} \right]^{2^j}$$

is obtained. Let us turn to comparisons. Since every node has either two or no offsprings, there will be always an even number of comparisons. Fix again $k = 2^j + i$, $0 \leqslant i \leqslant 2^j - 1$, and let $t = 2r$ with $r \geqslant 1$. Then

$$|\{g; 2d(k,g) + s(g) = 2r\}| = A_1 + A_2,$$

where

$$A_1 := |\{g; 2d(k,g) + s(g) = 2r, g \text{ is a leaf}\}|,$$
$$A_2 := |\{g; 2d(k,g) + s(g) = 2r, g \text{ is an interior node}\}|.$$

Hence

$$A_1 = 0 \qquad \text{if} \quad r \neq n-j,$$
$$= 2^r \qquad \text{if} \quad r = n-j,$$

and

$$A_2 = 2^{r-1} \qquad \text{if} \quad r \leqslant n-j$$
$$= 0 \qquad \text{otherwise.}$$

This yields

$$\mathscr{V}_{2j+i}(z) = \frac{1}{2^{n+1-j}-1} \cdot \frac{2^{n-j} z^{2(n-j)}(3z^2-1) - z^2}{2z^2-1},$$

thus the generating function $\mathscr{V}(z)$ in question equals

$$\prod_{j=0}^{n-1} \left[\frac{1}{2^{n+1-j}-1} \cdot \frac{(2z^2)^{n-j}(3z^2-1) - z^2}{2z^2-1} \right]^{2^j}.$$

Let us evaluate the corresponding expectations and variances in this interesting special case. Abbreviating

$$t_j(z) := \frac{(2z)^{j+1}-1}{(2^{j+1}-1)(2z-1)},$$

$\mathscr{E}$ may be written as

$$\mathscr{E}(z) = \prod_{j=1}^{n} (t_j(z))^{2^{n-j}}.$$

Taking logarithmic derivatives and using

$$\mathscr{E}(1) = t_j(1) = 1,$$

the expected number of interchanges may be written as

$$\mathscr{E}'(1) = \sum_{j=1}^{n} 2^{n-j} t_j'(1).$$

In a similar way,

$$\mathscr{V}'(1) = \sum_{j=1}^{n} 2^{n-j} v_j'(1)$$

gives the expected number of comparisons, where

$$v_j(z) := \frac{(2z^2)^j(3z^2-1) - z^2}{(2^{j+1}+1)(2z^2-1)}$$

is set for abbreviation.

An easy computation shows

$$t_j'(1) = \frac{(j+1)2^{j+1}}{2^{j+1}-1} - 2$$

and

$$v_j'(1) = \frac{(2j-1)2^{j+1}+2}{2^{j+1}-1},$$

so

$$\mathscr{E}'(1) = 2^{n+1} \sum_{j=2}^{n+1} \frac{1}{2^j - 1} - 2(2^n - 1) = 2^{n+1}(y_{n+1} - 2) + 2$$

with

$$y_n := \sum_{j=1}^{n} \frac{j}{2^j - 1}.$$

Setting

$$x_n := \sum_{j=1}^{n} \frac{1}{2^j - 1},$$

one obtains similarly

$$\mathscr{V}'(1) = 2^{n+1}(2y_{n+1} - x_{n+1} - 2) + 2.$$

Hence in order to obtain an asymptotic estimate for the expectations in question, we need

LEMMA 3.2. *As* $n \to \infty$,

$$x_n = \alpha_1 - \frac{1}{2^n} - \frac{1}{3 \cdot 2^{2n}} + \sigma\left(\frac{1}{2^{2n}}\right),$$

$$y_n = \alpha_1 + \alpha_2 - \frac{n}{2^n} - \frac{n}{3 \cdot 2^{2n}} + \sigma\left(\frac{n}{2^{2n}}\right),$$

where

$$\alpha_i := \sum_{j=1}^{\infty} \frac{1}{(2^j - 1)^i}.$$

Proof. We will obtain the asymptotic expansions in question from the generating functions for (x_n) and (y_n) by what is known as Darboux's method (Greene and Knuth, 1982, 4.3.1).

(1) The generating function for (x_n) is

$$\mathcal{F}(z) := \frac{1}{1-z} \sum_{k=1}^{\infty} \frac{z}{2^k - z}.$$

$\mathcal{F}$ has simple poles in 2^i, $i \geqslant 0$, and from Darboux's theorem we obtain as a first approximation

$$x_n = \alpha_1 + \sigma(1) \qquad \text{as} \quad n \to \infty.$$

Since $(z) - \alpha_1/(1-z)$ is regular in $|z| < 2$, this may be improved to

$$x_n = \alpha_1 - \frac{1}{2^n} + \sigma\left(\frac{1}{2^n}\right)$$

and finally to

$$x_n = \alpha_1 - \frac{1}{2^n} - \frac{1}{3 \cdot 2^{2n}} + \sigma\left(\frac{1}{2^{2n}}\right).$$

(2) Since

$$\mathcal{F}_0(z) := \sum_{k=1}^{\infty} \frac{z}{2^k - z}$$

is the generating function of

$$\left(\frac{1}{2^k - 1}\right)_{k \geqslant 1},$$

$z \cdot \mathscr{F}_0'(z)$ will be the generating function for

$$\left(\frac{k}{2^k-1}\right)_{k\geqslant 1},$$

consequently,

$$\mathscr{G}(z) := \frac{x}{1-x} \sum_{k=1}^{\infty} \frac{2^k}{(2^k-x)^2}$$

is the generating function for $(y_n)_{n\geqslant 1}$. $\mathscr{G}$ has a simple pole at 1 and double poles at 2^k, $k \geqslant 1$, so from Darboux's theorem we get as a first approximation

$$y_n = \alpha_1 + \alpha_2 + o(1)$$

from the simple pole at $z = 1$.

Since

$$\mathscr{G}(z) - \frac{\alpha_1+\alpha_2}{1-z}$$

is regular in $|z| < 2$ (this can be seen from the partial fraction expansion

$$\frac{2^k \cdot z}{(1-z)(z-2^k)^2} = \frac{2^k}{(2^k-1)^2(z-2^k)} - \frac{2^{2k}}{(2^k-1)(z-2^k)^2} + \frac{2^k}{(2^k-1)^2(1-z)}\Bigg),$$

this first approximation may be improved to

$$y_n = \alpha_1 + \alpha_2 - \frac{n}{2^n} + o\left(\frac{n}{2^n}\right),$$

and, by a similar argument applied to

$$\mathscr{G}(z) - \frac{\alpha_1+\alpha_2}{1-z} + \sum_{n=1}^{\infty} \frac{nz^n}{2^n} = \mathscr{G}(z) - \frac{\alpha_1+\alpha_2}{1-z} + \frac{4}{(z-2)^2},$$

we finally get the claimed expansion. ∎

Together with the expressions for $\mathscr{E}'(1)$ and $\mathscr{V}'(1)$ from above these expansions yield

PROPOSITION 3.3. *If* $N = 2^{n+1} - 1$, *Floyd's algorithm for heap construction requires*

$$(\alpha_1 + \alpha_2 - 2)N - n + \alpha_1 + \alpha_2 - 1 - \frac{n}{3N} + \sigma\left(\frac{n}{N}\right)$$

interchanges, and

$$(\alpha_1 + 2\alpha_2 - 2)N - 2n + \alpha_1 + 2\alpha_2 - 1 - \frac{2n}{3N} + \sigma\left(\frac{n}{N}\right)$$

comparisons, as $n \to \infty$.

Numerically, we get

$$\alpha_1 = 1.6066951...,$$

and

$$\alpha_2 = 1.1373387...,$$

so the coefficients for the leading term are

$$\alpha_1 + \alpha_2 - 2 = 0.7440338...,$$

and

$$\alpha_1 - 2\alpha_2 - 2 = 1.8813726....$$

Let us have a look at the variances of the quantities in question. The variance for interchanges equals

$$\operatorname{var}(\mathscr{E}) := \mathscr{E}''(1) + \mathscr{E}'(1) - (\mathscr{E}'(1))^2.$$

From

$$\mathscr{E}'(z) = \mathscr{E}(z) \sum_{j=1}^{n} 2^{n-j} \frac{t_j'(z)}{t_j(z)}$$

and

$$\mathscr{E}(1) = t_j(1) = 1$$

it is easily derived that

$$\operatorname{var}(\mathscr{E}) = \sum_{j=1}^{n} 2^{n-j}(t_j''(1) + t_j'(1) - (t_j'(1))^2),$$

similarly,

$$\operatorname{var}(\mathscr{V}) = \sum_{j=1}^{n} 2^{n-j}(v_j''(1) + v_j'(1) - (v_j'(1))^2).$$

Since

$$t_j''(1) = \frac{2(j-4)(j+1)2^{j+1}}{2^{j+1}-1} + 8,$$

$$v_j''(1) = \frac{j(2j-3)2^{j+2}-7}{2^{j+1}-1} + 7,$$

we get after some simplification

$$\operatorname{var}(\mathscr{E}) = 2^{n+1} \sum_{j=1}^{n} \left(2^{-j} - \frac{(j+1)^2}{(2^{j+1}-1)^2}\right) = 2^{n+2} - 2^{n+1}\hat{y}_{n+1} - 2$$

and

$$\begin{aligned}\operatorname{var}(\mathscr{V}) &= 2^{n+1} \sum_{j=1}^{n} \left(4 \cdot 2^{-k} - \frac{3}{2^{k+1}-1} - \frac{4k^2+4k+1}{(2^{k+1}-1)^2}\right) \\ &= 2^n(8\hat{x}_{n+1} - \hat{y}_{n+1} - 2y_{n+1} - 6x_{n+1} + 9) - 8,\end{aligned}$$

where

$$\hat{x}_n := \sum_{k=1}^{n} \frac{k}{(2^k-1)^2},$$

$$\hat{y}_n := \sum_{k=1}^{n} \frac{k^2}{(2^k-1)^2}.$$

In order to derive an asymptotic expansion for these variances, we need

LEMMA 3.4. *Asymptotically,*

$$\hat{x}_n = \alpha_3 - \frac{n}{3 \cdot 2^{2n}} + \sigma\left(\frac{n}{2^{2n}}\right)$$

and

$$\hat{y}_n = \alpha_4 - \frac{n^2}{3 \cdot 2^{2n}} + \sigma\left(\frac{n^2}{2^{2n}}\right) \qquad \text{as} \quad n \to \infty,$$

where

$$\alpha_3 := \sum_{k=1}^{\infty} \frac{k2^{k+1}}{(2^{k+1}-1)^2},$$

$$\alpha_4 := \sum_{k=1}^{\infty} \frac{k2^{k+1}(2^{k+1}+1)}{(2^{k+1}-1)^3}.$$

Proof. The expansions in question are again derived from the generating functions for the sequences. Since

$$\mathcal{G}_0(z) := \sum_{k=1}^{\infty} \frac{k \cdot z}{2^{k+1} - z}$$

is the generating function for $(1/(2^{n-1})^2)_{n \geq 1}$, we may derive, using the simple rules for manipulating generating functions (Knuth 1973a, 1.2.9), that

$$\mathcal{G}_1(z) := \frac{z\mathcal{G}_0'(z)}{1-z} = \frac{z}{1-z} \sum_{k=1}^{\infty} \frac{k \cdot 2^{k+1}}{(2^{k+1} - z)^2}$$

is the generating function for $(\hat{x}_n)$. Similarly,

$$\mathcal{G}_2(z) := \frac{z}{1-z} \sum_{k=1}^{\infty} \frac{k \cdot 2^{k+1}(2^{k+1}+1)}{(2^{k+1} - z)^3}$$

may be seen the generating function for $(\hat{y}_n)$. The expansions now may be derived from Darboux's theorem in exactly the same way as before. ∎

Before proceeding, a comment about the constants $\alpha_1, \ldots, \alpha_4$ is in order. These constants may be obtained from the basic hypergeometric series

$$\phi(x, y, q) := 1 + \sum_{k=1}^{\infty} \frac{x^k}{1 - y^k q}$$

in the following manner:

$$\alpha_1 = \phi\left(\frac{1}{2}, \frac{1}{2}, 1\right) - 1,$$

$$\alpha_2 = \frac{\partial}{\partial q} \phi\left(\frac{1}{2}, \frac{1}{2}, q\right)\Big|_{q=1},$$

$$\alpha_3 = \frac{1}{2} \frac{\partial^2}{\partial x \, \partial q} \phi\left(x, \frac{1}{2}, q\right)\Big|_{x=1, q=1/2},$$

and finally

$$\alpha_4 = \frac{1}{4} \frac{\partial^2}{\partial y \, \partial q} \phi(1, y, q)|_{y=1/2, q=1/2}$$

(it is not difficult to see that the partial derivatives at the points in question may be taken). Numerically, we have

$$\alpha_3 = 1.3085437\ldots,$$

$$\alpha_4 = 1.7387828\ldots.$$

Plugging these asymptotic expansions into the expressions found for the variances above, and doing some simplification yields

PROPOSITION 3.5. *If* $N = 2^{n+1} - 1$, *Floyd's algorithm to construct heaps has the following variances, as* $n \to \infty$:

(a) *for interchanges*: $(2 - \alpha_4)N - \alpha_4 + n^2/3N + o(n^2/2^n)$

(b) *for comparisons*: $(4\alpha_3 - 4\alpha_1 - \alpha_2 - \alpha_4/2 + \frac{9}{2})N + 4\alpha_3 - 4\alpha_1 - \alpha_2 - \alpha_4/2 + \frac{9}{2} + n + n^2/6N + o(n^2/2^n)$.

Numerically, the leading term has the following coefficient: 0.2612171..., (interchanges), 1.3006643 (comparisons).

These variances are linear; it would be interesting to have, at least in case $N = 2^{n+1} - 1$, an asymptotic expansion for the mth moment for interchanges, both as $m \to \infty$, and as $n \to \infty$. This could possibly follow the lines of (Doberkat, 1982, Prodinger, 1984).

Let us have a look at the general situation and derive the corresponding generating functions. In order to do this, we have to distinguish between special and nonspecial nodes in the tree. Suppose the node k is a nonspecial node on level j, then the considerations above imply that

$$\mathscr{E}_k(z) = (2^{n+1-j} - 1)^{-1} \frac{(2z)^{n+1-j} - 1}{2z - 1}, \qquad k \text{ is a left node,}$$

$$= (2^{n-j} - 1)^{-1} \frac{(2z)^{n-j} - 1}{2z - 1}, \qquad k \text{ is a right node,}$$

and similar formulae hold for $\mathscr{V}_k(z)$. However, if k is a special node, $\mathscr{E}_k(z)$ and $\mathscr{V}_k(z)$ do not look so regular. Suppose $k = (1b_{n-1}, \ldots, b_j)_2$, then

$$\gamma(k) = (1b_{j-1}, \ldots, b_0)_2.$$

Since the distance from k to N is j, we see that with

$$A_r := |\{t; t \text{ is the subtree rooted at } k,\ d(t, k) = r\}|,$$

the following holds

$$A_r = 2^r \qquad \text{if} \quad 0 \leqslant r < j,$$
$$= (b_{j-1} \cdots b_0)_2 + 1, \qquad \text{if} \quad r = j.$$

Hence

$$\mathscr{E}_k(z) = \gamma(k)^{-1} \left[\frac{2^{k+1} z^k (1 - z) - 1}{2z - 1} + (\gamma(k) + 1) z^j \right].$$

Abbreviating $(1b_{n-1}\cdots b_j)_2$ by $\sigma(N,j)$, and $(1b_{j-1}\cdots b_0)_2$ by $\tau(N,j)$, the probability generating function for interchanges has according to Theorem 3.1 the following form:

$$\begin{aligned}\mathscr{E}(z) &= \prod_{j=1}^{n}\left[\frac{1}{2^{n+1-j}-1}\,\frac{(2z)^{n+1-j}-1}{2z-1}\right]^{\sigma(N,n-j)-2^j}\\ &= \prod_{j=1}^{n-1}\left[\frac{1}{2^{n-j}-1}\,\frac{(2z)^{n-j}-1}{2z-1}\right]^{2^{j+1}-1-\sigma(N,n-j)}\\ &= \prod_{j=0}^{n}\frac{1}{\tau(N,j)}\left[\frac{2^{j+1}z^j(1-z)-1}{2z-1}+(\tau(N,j)+1)z^j\right].\end{aligned}$$

Let us turn to the generating function for comparisons, and again let $k=(1b_{n-1}\cdots b_j)_2$ be a special node. Suppose t is a node in the subtree rooted at k, then we have to have a look at $f(k,t)$, the number of comparisons which are done in case the label of k percolates the tree and labels node t. It is quite immediate that the following holds:

$$\begin{array}{lll} f(k,t)=2j & \text{if} & d(k,t)=j,\ t\neq N,\\ \quad 2(j-1) & \text{if} & d(k,t)=j-1 \text{ and } t \text{ is a leaf},\\ \quad 2(j-1)+b_0+1 & \text{if} & t=\lfloor N/2\rfloor \text{ or } t=N,\\ \quad 2(d(k,t)+1) & \text{otherwise} & \end{array}$$

Thus if

$$A_r := |\{t;\ t \text{ is in the subtree rooted at } k, f(k,t)=r\}|,$$

these considerations yield

$$\begin{array}{lll} A_r=2^{l-1} & \text{if} & r=2l,\ 1\leqslant l\leqslant j-2,\\ \quad 3\cdot 2^{j-2}-1-(b_{j-1}\cdots b_1)_2 & \text{if} & r=2(j-1),\\ \quad 2(1-b_0) & \text{if} & r=2j-1,\\ \quad 3(b_{j-1}\cdots b_1)_2+3b_0 & \text{if} & r=2j.\end{array}$$

Consider, e.g., the case $r=2(j-1)$: $f(k,t)$ will be equal to $2(j-1)$ if

(a) t is a leaf to the right of $\lfloor N/2\rfloor$ or

(b) t is an interior node with $d(k,t)=j-2$.

There are $2^{j-1}-1-(b_{j-1}\cdots b_1)_2$ leaves to the right of $\lfloor N/2\rfloor$, and 2^{j-2} interior nodes the distance of which to k is $j-2$. Hence

$$\begin{aligned}\mathscr{V}_k(z) = \frac{1}{(1b_{j-1}\cdots b_0)_2}\Bigg[&z^2\,\frac{2^{j-z}z^{2(j-2)}-1}{2z^2-1}\\ &+z^{2(j-1)}((b_{j-1}\cdots b_1)_2(3z^2-1)+b_0(3z^2-2z)+2z+3\cdot 2^{j-2}-1)\Bigg],\end{aligned}$$

and this gives the following expression for the probability generating function for comparisons:

$$\begin{aligned}\mathscr{V}_k(z) &= \prod_{j=1}^{n}\left[\frac{1}{2^{n+1-j}-1}\,\frac{(2z^2)^{n-j}(3z^2-1)-z^2}{2z^2-1}\right]^{\sigma(N,n-j)-2^j}\\ &= \prod_{j=1}^{n-1}\left[\frac{1}{2^{n-j}-1}\,\frac{(2z^2)^{n-1-j}(3z^2-1)-z^2}{2z-1}\right]^{2^{j+1}-1-\sigma(N,n-j)}\\ &= \prod_{j=1}^{n}\frac{1}{\tau(N,j)}\left[\frac{z^2(2z^2)^{j-2}-z^2}{2z^2-1}\right.\\ &\quad + z^{2(j-1)}((b_{j-1}\cdots b_1)_2(3z^2-1)\\ &\quad \left.+ b_0(3z^2-2z)+2z+3\cdot 2^{j-2}-1)\right].\end{aligned}$$

Evaluating $\mathscr{E}'(1)$ and $\mathscr{V}'(1)$, respectively, yields

THEOREM 3.3. *Floyd's algorithm requires*

$$(\alpha_1+\alpha_2-2)N+\mathscr{O}(\log N)$$

interchanges and

$$(\alpha_1+2\alpha_2-2)N+\mathscr{O}(\log N)$$

comparisons on the average. ∎

It is possible to express the corresponding $\mathscr{O}(\log N)$ terms by means of the binary expansion of N; but since this results in rather clumsy and awkward expressions, only the leading term of the respective asymptotic expansions is given here.

ACKNOWLEDGMENTS

Some calculations were done with MACSYMA, developed by the Mathlab Group at MIT which is supported in part by the United States Energy Research and Development Administration under Contract Number E(11-1)-3070 and by the National Aeronautics and Space Administration under Grant NSG 1323.

RECEIVED: September 2, 1982; ACCEPTED: July 25, 1984

REFERENCES

AHO, A. V., HOPCROFT, J. E., AND ULLMAN, J. D. (1974), "Design and Analysis of Algorithms," Addison–Wesley, Reading, Mass.

DOBERKAT, E. E. (1980), Some observations on the average performance of heapsort, *in* "Proc. 21st Ann. IEEE Sympos. Math. Found. Comput. Sci." IEEE Computer Society, Los Angeles, 229–237.

DOBERKAT, E. E. (1982), Asymptotic estimates for higher moments of the expected behavior of straight insertion sort, *Inform. Process. Lett.* **14**, 179–182.

DOBERKAT, E. E. (1983), Continuous models that are equivalent to randomness for the analysis of many sorting algorithms, *Computing* **31**, 11–31.

FLOYD, R. W. (1964), Algorithm 245, treesort 3, *Comm. ACM* **7**, 701.

GREENE, D. H. AND KNUTH, D. E. (1981), "Mathematics for the Analysis of Algorithms," Birkhauser, Boston.

KNUTH, D. E. (1973a), "The Art of Computer Programming Vol. I: Fundamental Algorithms," (2nd ed.), Addison–Wesley, Reading, Mass.

KNUTH, D. E. (1973b), "The Art of Computer Programming, Vol. III: Sorting and Searching," Addison–Wesley, Reading, Mass.

PRODINGER, H. (1984), On a question by Doberkat about the higher moments of the expected behavior of straight insertion sort, *Inform. Process. Lett.*, in press.

RUDIN, W. (1974), "Real and Complex Analysis." McGraw–Hill, New York (Second Edition).