

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,200

Open access books available

169,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Introductory Chapter: Search Algorithm - Essence of Optimization

Dinesh G. Harkut

1. Introduction

Every time you hit the search button on Google, the search engine sifts through thousands of searches, if not millions of web pages, to spit out the content we are seeking in a fraction of a second. It is an algorithm – a set of mathematical rules embedded in the software, which makes all this possible. In fact, every time we enlist for a unique identity social security/Aadhaar number, access an automated teller machine, book train or air tickets, or buy merchandise online, we are indirectly expanding the scope and range of algorithms, a mathematical concept whose roots date back to 600 AD with the invention of the decimal system.

Algorithms are nothing but the logically group of instructions aimed at solving a problem or completing a task. Recipes are algorithms like math equations. Computer code is algorithmic. Algorithms are aimed at optimizing everything. Mathematical algorithms include fundamental methods from arithmetic and numerical analysis, which in turn manipulate the data through addition, multiplication of integers, polynomials, and matrices, and may be used for solving a large variety of mathematical problems which arise in many contexts: solution of simultaneous equations, data fitting and integration, and random number generation. Here, the main emphasis is on algorithmic aspects of the methods, rather than the mathematical basis.

Whenever we use a computer, laptop, phone, or a mileage calculator in a car, we are using algorithms, and we may call it programs, or software packages, or apps. They can make things easier, save lives, and surmount disorder. To discuss the effects of technology-enabled assistance in human lives, algorithms are a useful artifact to begin with. Algorithms have penetrated in every aspect of human life and provide a better standard against which to compare human cognition itself. It becomes the new arbiters of human decision-making in almost any area we can imagine like which movie to watch to which house to buy to self-driving cars. Biometrics refer to identifying human being by certain physical features like fingerprints or iris scans. Biometrics-based social security identity card or Aadhaar card, in Indian context which evolved as the India's universal identity card, in turn uses an algorithm to store and retrieve fingerprints and iris scans. Computer scientists have devised algorithms that can analyze a given thumbprint and match it against a database. Because of the overdependence of human beings on computer, indirectly only the algorithms determine whether one gets a job or one get into college or get an apartment; moreover, their work goes largely unnoticed. Algorithms are behind many routines works, but they are still significant decision-making tools in everyone's life.

Deloitte Global predicted that, in time to come, more than 80% of the world's largest software enterprise companies will have cognitive technologies, mediated by algorithms integrated into their products. Algorithms with the perseverance and ubiquity of insects shall automate the processes that are used to require human intervention and rational. These can now achieve basic processes of measuring, monitoring, seeing, or even counting. Our vehicle can guide us where to slow down. Our television sets can now advocate which movies to watch. A grocery can recommend a healthy amalgamation of foods and vegetables for lunch/dinner. Alexa/Siri reminds us important events or anniversaries of dear ones. The overall impact of pervasive algorithms is very hard to calculate because the presence of algorithms in every walk of life, everyday processes, and transactions is now so great and is mostly hidden from public view.

Algorithms are making enormously significant pronouncements in our society in almost every walk of life, ranging from welfare benefits to medicine to transportation to criminal justice and beyond. The ever-increasing assortment and investigation of data and the resulting application of this information can decrease poverty, cure diseases effectively, bring apt resolutions to mankind, places where need is utmost, and dispel epochs of prejudgment, illogical suppositions, vicious practice, and obliviousness of all kinds. Algorithms are now redefining how we think, what we know, and what we think. Algorithms are a black box and are invisible pieces of code that tell a computer how to accomplish a specific task. An algorithm directs the computer what to do in order to produce a certain desired outcome. Every time you do search on internet through any search engine like Google or look at your Facebook feeds or use GPS navigation in your car, you are directly or indirectly interacting with an algorithm. Individuals often demonstrate greater trust on assistance from algorithms compared to non-algorithmic assistance, displaying algorithmic obligation. Counting on algorithms for analytical tasks is typically beneficial. Even simple algorithms, such as weighting all variables equally, can outclass humanoid prediction. Algorithms have begun to intrude on tasks conventionally earmarked for human judgment and are progressively proficient of performing well in innovative and tough tasks. Moreover, at the same time, societal impact, through social media, personal networks, or online assessments and reviews, is one of the most compelling forces affecting individual decision-making.

In short, algorithms are the core entity of the internet, and they manage and run the internet and all online activities like financial transactions, crypto/stock trading, searching, customized browsing, data manipulation, etc. Email knows the destination address and thus knows where to go thanks to the underlying algorithms. Moreover, smartphone mobile apps are nothing but algorithms. Computer and video games are algorithmic storytelling. Book or movie recommendation, online dating, leisure\travel web portals, and so on would not function properly and efficiently without algorithms. Artificial intelligence (AI) is nothing but algorithms. GPS mapping systems make use of algorithms to get entity from location X to location Y. Every single piece of the object people see on social media is brought to them by means of algorithms. Moreover, everything people do in everyday life and see on the web is an outcome of some algorithm or other. Every time we sort or arrange data in a worksheet, algorithms plays vital role, and almost every financial transaction is accomplished today by algorithms. Algorithms make every electronic gadget to respond to voice commands, organize and sort photos, recognize and identify faces, and build and drive automobile cars. Hacking, cyberattacks, and cryptographic code-breaking exploit algorithms to the next level. Algorithms are often sophisticated, elegant, and amazingly useful tools used to accomplish various categories of tasks. They are mostly hidden and invisible aids, enhancing and augmenting human lives in increasingly efficient ways. Algorithms will continue to face

the ever-increasing impact over the next few decades, influencing people's work and personal lives and the ways they interact with information, organizations like health care service providers, not-for-profit/government institutions, banking/financial sector, retailers/traders, education, media corporates houses and entertainment industry, and each other. The hope is that algorithms will help people swiftly and impartially perform the tasks and get the desired products, information, and services. The major apprehension is that algorithms can deliberately or unconsciously create discrimination and thus enable social engineering to create biased narrative and have other harmful societal impacts.

The term algorithm which finds application in computer science is universally used to describe problem-solving methods that help for the implementation of computer programs. Mostly, algorithms involve complicated methods of manipulating and organizing the data involved in the computation. Basically, they involved the manipulation of data based on some mathematical model and which typically find applications of: searching, sorting, string processing, geometric algorithms, graph algorithms, genetic algorithms (GAs), neural network, etc.

2. Searching

Searching is a method for finding certain things in given data/files which are of vital importance. There are different categories of search methods: basic and advanced, like one using trees and digital key transformations, including balanced trees, hashing binary search trees, and digital search trees, and trying different methods appropriate for different types of files. These methods in turn are related to each other and possess much resemblance with sorting methods.

3. Sorting

Sorting is a method of rearranging files in the order that are covered in some depth due to their fundamental importance. A large variety of methods were developed, described, and compared. Algorithms including priority queues, selection, merging, and several related problems are created. Some of these algorithms are used as the basis for other more complex algorithms.

4. String processing

String processing algorithms include a range of methods for dealing with (long) sequences of characters. String searching basically leads to pattern matching which in turn leads to parsing. File compression techniques and cryptology are also part of advanced string processing applications.

5. Geometric algorithms

Geometric algorithms encompass an assortment of procedures for resolving problems by involving points and lines along with other simple geometric objects. Moreover, it also finds application of ideas from several mathematical disciplines like algebra, combinatorics, topology, and differential geometry. During the last two decades, most of the geometric applications like CAD/CAM, computer graphics, VLSI design, molecular biology, robotics, GIS, spatial databases, sensor

networks, machine learning, and scientific computing become the motivation for computational geometry to evolve as a full-fledged discipline of theoretical computer science. Some typical high-end applications of geometric algorithms includes: optimal airspace design, air traffic controller's traffic balancing and automatic sectorization of airspace, wireless sensor networks design, analysis of 2D electrophoresis gels, prediction of resilience, and recovery of damage in neural networks.

6. Graph algorithms

Graphs algorithms are the natural way to understand connection between the linked data and thus reveal the relationships in data. Exploring and tracking these interlinking connections divulge new insights and influence and facilitate to analyze each data point as part of a bigger picture. Graph algorithms are useful in a variety of complex difficult structures and reveal difficult-to-find patterns ranging from finding bottlenecks, susceptibilities to detect communities, fraud rings, improving machine learning predictions to predict the spread of disease or ideas, and thus enable us to leverage relationship within data to devise intelligent solution to enhance the effectiveness of machine learning models. Graph algorithms types such as exact or approximated, static or dynamic, distributed or centralized, deterministic or randomized, and matching and network flow, minimal spanning tree, and shortest path are some of the general approaches developed for searching in the graph.

7. Genetic algorithms

A genetic algorithm (GA) is a search-based heuristic algorithm used for solving optimization problems in machine learning that is based on genetics and natural selection. GA is a subset of evolutionary algorithms which is based on behavior of chromosomes and their genetic structures which uses evolutionary generational cycle starting from initialization, fitness assignment, selection, reproduction, replacement, and termination, to yield high-quality solutions. GA makes use of various operations which enhance or replace the population to deliver a better-quality suitable solution and get inspiration from evolution and natural selection. Through the process of natural selection, organisms regulate to augment their likelihoods for endurance in a given situation. Eventually, incompetent elements perish from the population, to be replaced by successful-solution descendants. Apart from the applications in multimodal optimization, DNA analysis, design of aircraft, and genetic algorithms are found to be efficient and cost-effective plan for tricky traveling salesman problem. It does not need derivative information and possess excellent parallel capabilities which refine and optimize the solution to the multiobjective problems, and discrete and continuous functions. The idea behind genetic algorithms is extremely alluring.

8. Neural networks

Neural networks are basically inspired by the biological nervous system or neural networks in the brain and basically parallel computing devices. It is one of the most emerging areas in data science which revolutionizes and eventually leads to tremendous growth of artificial intelligence, machine learning, and deep learning and basically designed to recognize patterns and extracting features that can

be fed to other relevant algorithms for further classifications and clustering. The elementary computational unit of a neural network is a neuron also called as node, and each node or neuron is linked with certain weight. This weight is assigned as per the relative importance of that particular neuron or node. Neural networks are simply weighted digraphs with neurons acting as vertices, and weights on edges denote the connection strength of the pair. Each node or neuron collects values or weights from other neurons and accordingly computes the output.

Gradient Decent, Conjugate Gradient, Newton's Method, and Quasi-Newton Method are some of the popular optimization algorithms which find applications in context with neural network. Speed and memory footprint of all these algorithms may vary, but an ultimate objective is to accomplish various intricate computational task faster than the traditional systems.

In neural network, training of the neurons is accomplished by appropriately modifying connection strength in response to training data. Most apt application of neural networks is forecasting and classification of applications, such as gene predication, optical character recognition, and stock market time series prediction.

Having discussed about all these different types of algorithms, one thing which is most apparent that, irrespective of types of algorithms, searching lies at the heart of all. Rather, it is the intrinsic part of all algorithms as it finds application one way or other, and it is one of the first things any algorithm designer should try in the quest for efficiency. Basically, sorting is directly based on searching, wherein we use to compare, i.e., search for specific pattern of string for comparison, and then rearrange those strings based on desired sorting pattern of ascending or descending. Searching can be used to illustrate most algorithm design paradigms. Data structure techniques, divide-and-conquer, randomization, and incremental construction all lead to popular sorting algorithms. Binary search and its variants are the essential divide-and-conquer algorithms. Depth-first and breadth-first search provide mechanisms to visit each edge and vertex of the given graph. Strategy represents the pursuit for the big picture, the framework around which we construct our path to the goal. Strategies are used to win the minor skirmishes we must fight along the way. They prove the basis of most simple and efficient graph algorithms. Simulated annealing is a simple and effective technique to efficiently obtain good but not optimal solutions to combinatorial search problems. Combinatorial search, improved with tree pruning techniques, can then be used to find the more optimal solution of small optimization problems. Ingenious pruning procedures can speed up combinatorial search to a remarkable extent. Proper trimming will have a greater impact on search time than any other factor. Historically, computers have spent more time searching and sorting than doing anything else all put together. A quarter of all mainframe cycles are spent in searching and sorting data. Although it is unclear whether this remains true on smaller computers, but still searching and sorting remain the most ubiquitous combinatorial algorithm problem in practice.

An important key to algorithm design is to use searching/sorting as a basic building block, because once a set of items is sorted, many other problems become easy. Consider the few of the applications like:

- **Closest pair** – In a given a set of n different numbers, suppose we are interested in finding the pairs of numbers having minimum or no deviation or minimal difference between them. One way to find the desired pairs in quickest possible way is to arrange these numbers in sorted order either in ascending or descending order. Once sorted, the closet pair of numbers shall spatially reside next to each other. Thus, merely a linear scan through the sorted list will complete the job.

- **Element uniqueness** – In a given set of n items or list of numbers, suppose we are interested in finding the occurrence of any duplicates, the most efficient and appropriate algorithm is to sort these item lists followed by simple linear scan. By verifying and checking, indirectly searching all adjacent pairs, duplicate item can be pointed out. This can be treated as one of the special cases of the closest-pair problem, wherein instead of finding minimal deviation, we are typically interested in finding the elements in the list with deviation or gap of difference of zero.
- **Frequency distribution** – In a given a set of n items, suppose we want to find the major number of times or count of appearance/occurrence of particular element, i.e., frequency of occurrence, arranged the given list in sorted order either in ascending or descending, scan the list from left to right and go on counting them, as all matching elements will be lumped together during sorting. To find out how frequently an arbitrary selected element k occurs, one can start by looking up k using binary search in a sorted array of keys. By scanning the list from left of this point until the element is not k and then moving to the right, we can find this count c in linear time, where c is the number of occurrences of k . The number of instances of k can be found in time by using binary search to look for the positions of both and where it is arbitrarily small, and then by taking the difference of these positions.
- **Selection** – In a given list of numbers in array, suppose one is interested in finding the k^{th} biggest or largest item, the desired k^{th} largest can be found in fix constant time. If the numbers in the arrays are arranged and placed in proper sorted order, we simply need to look for the k^{th} position in the array, as the median element appears in the $(n/2)^{\text{nd}}$ position in sorted order list.
- **Convex hulls** – In a given set of n points in two dimensions, suppose we want to find the polygon having smallest area such that it encompasses all the points. Here, the convex hull is which is just like a rubber band stretched over the points in the given plane and then released gives a nice representation of the shape of the points and is the most important building block for more sophisticated geometric algorithms. Moreover, if the given points are arranged in sorted order in either of the coordinate, the points can be inserted either from left to right or bottom to top, respectively, into the hull. Consider the case that points are sorted on x coordinate, as the rightmost point is always lies on the edge, and it will be inserted into the hull. However, adding new rightmost point may cause others point to be deleted, and we can swiftly identify such points as they fall inside the polygon formed by adding these new points. Thus, points to be deleted are the neighbors of the preceding point we injected, so they will be easy to identy. The total time is linear after the sorting has been done. Although some of these issues can be solved in linear time using more sophisticated algorithms, sorting which is incidentally based on searching at core provides quick and easy solutions to all of these problems. It is a rare application whose time complexity is such that sorting proves to be the bottleneck, especially a bottleneck that could have otherwise been removed using clever algorithmics.

Thus, searching/sorting is the most thoroughly studied problem in computer science. Literally, loads of diverse algorithms are known, most of which have some advantages over all other algorithms in certain circumstances. Most of the stimulating concepts used in the design of algorithms appear in the context of searching/sorting such as data structures, divide-and-conquer, and randomized algorithms.

Searching/sorting is a natural laboratory for studying basic algorithm design paradigms, since many useful techniques lead to interesting searching/sorting algorithms. Searching/sorting thus becomes the basic building block around which many other algorithms are built. By understanding searching thoroughly and devising effective searching techniques, we can obtain an amazing amount of power to solve other problems and thus will become the essence for optimizations.

IntechOpen

IntechOpen

Author details

Dinesh G. Harkut
Prof. Ram Meghe College of Engineering and Management Badnera, India

*Address all correspondence to: dg.harkut@gmail.com

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 