1. Explain the concept of algorithm in your own words.

Algorithms for the same problem can be based on very different ideas and can solve the problem with dramatically different speeds.

An ***algorithm*** is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.

1. What is correctness of an algorithm?

An algorithm Is called totally correct for the given specification if and only if for any correct input data it: 1 stops and 2 returns correct output

3. What is the difference between partially and fully correct algorithm?

The **difference between partial** correctness and total correctness is that a **totally correct algorithm** requires the **algorithm** to terminate, while a **partially correct algorithm** is one that doesn't have a terminating function but produces a **correct** result if halted.

4. What is termination proof?

A **termination proof** is a type of mathematical **proof** that plays a critical role in formal verification because total correctness of an algorithm depends on **termination**. A simple, general method for constructing **termination proofs** involves associating a measure with each step of an algorithm.

5. What are two main qualities to consider when analyzing the efficiency of an algorithm?

Worst case, Best case.

6. Which of the following formulas can be considered an algorithm for computing the area of a triangle whose side lengths are given positive numbers a, b, and c?

a. S = √p(p − a)(p − b)(p − c), where p = (a + b + c)⁄2

7. What are some of the problem types in algorithmics?

Sorting, Searching, String Processing, Graph Problems, Combinatorial Problems, Geometric Problems, Numerical Problems.

8. Give some examples of graph problems.

Some graph problems are computationally very hard; the most well-known examples are the traveling salesman problem and the graph-coloring problem. The ***traveling salesman problem (TSP)*** is the problem of finding the shortest tour through n cities that visits every city exactly once. In addition to obvious appli- cations involving route planning, it arises in such modern applications as circuit board and VLSI chip fabrication, X-ray crystallography, and genetic engineer- ing. The ***graph-coloring problem*** seeks to assign the smallest number of colors to the vertices of a graph so that no two adjacent vertices are the same color. This problem arises in several applications, such as event scheduling: if the events are represented by vertices that are connected by an edge if and only if the correspond- ing events cannot be scheduled at the same time, a solution to the graph-coloring problem yields an optimal schedule.

9. Give some examples of string processing problems.

One particular problem—that of searching for a given word in a text—has attracted special attention from researchers. They call it ***string matching***. Several algorithms that exploit the special nature of this type of searching have been invented.

10. What is traveling salesman problem?

The ***traveling salesman problem (TSP)*** has been intriguing researchers for the last 150 years by its seemingly simple formulation, important applications, and interesting connections to other combinatorial problems. In layman’s terms, the problem asks to find the shortest tour through a given set of n cities that visits each city exactly once before returning to the city where it started. The problem can be conveniently modeled by a weighted graph, with the graph’s vertices representing the cities and the edge weights specifying the distances. Then the problem can be stated as the problem of finding the shortest ***Hamiltonian circuit*** of the graph. (A Hamiltonian circuit is defined as a cycle that passes through all the vertices of the graph exactly once. It is named after the Irish mathematician Sir William Rowan Hamilton (1805–1865), who became interested in such cycles as an application of his algebraic discoveries.)

11. What are some real-world applications of string matching algorithms?

String Matching is the classical and existing problem, despite the fact that the real world aspects belonging to the research field of computer science. In this domain one or several strings called " Pattern " is to be searched within a well-built string or " Text ". String matching strategies or algorithms provide key role in various real world problems or applications. A few of its imperative applications are Spell Checkers, Spam Filters, Intrusion Detection System, Search Engines, Plagiarism Detection, Bioinformatics, Digital Forensics and Information Retrieval Systems etc. This paper is inclusive of analyzing nutshells about string matching along with its long-ago contributory details in an assortment of real world applications.

12 What is the worst-case and best-case efficiency of an algorithm?

Worst Case

In computer science, the **worst**-**case complexity** (usually denoted in asymptotic notation) measures the resources (e.g. running time, memory) an **algorithm** requires in the **worst**-**case**. It gives an upper bound on the resources required by the **algorithm**.

Best Case

The term *best-case performance* is used in computer science to describe an algorithm's behavior under optimal conditions. For example, the best case for a simple linear search on a list occurs when the desired element is the first element of the list.

Development and choice of algorithms is rarely based on best-case performance: most academic and commercial enterprises are more interested in improving Average-case complexity and worst-case performance. Algorithms may also be trivially modified to have good best-case running time by hard-coding solutions to a finite set of inputs, making the measure almost meaningless.

13. For each of the following functions, indicate how much the function’s value will change if its argument is increased fourfold.

a. log2 n

b. √n

c. n

d. n^2

e. n^3

f. 2^n

1. Log2 4n − log2n = (log2 4 + log2 n) − log2 n = 2.

b.√4n/ √n = 2.

c.4n/n = 4.

d. (4n)2/n2 = 42.

e. (4n)3/ n3 = 43.

f.24n/2n = 23n = (2n)3.

14. For each of the following algorithms, indicate (i) a natural size metric for its inputs, (ii) its basic operation, and (iii) whether the basic operation count can be different for inputs of the same size:

a. computing the sum of n numbers

b. computing n!

c. finding the largest element in a list of n numbers

a. (i) n; (ii) addition of two numbers; (iii) no

b. (i) the magnitude of n, i.e., the number of bits in its binary representation; (ii) multiplication of two integers; (iii) no

c. (i) n; (ii) comparison of two numbers; (iii) no (for the standard list scanning algorithm

15. Imagine that after washing 5 distinct pairs of socks, you discover that two socks are missing. Of course, you would like to have the largest number of complete pairs remaining. What are the best-case and the worst-case scenarios? Assuming that the probability of disappearance for each of the 10 socks is the same, find the probability of the best-case scenario; the probability of the worst-case scenario.

**Part A: The probability on the best case scenario**

The best case scenario results when the two missing socks are of the same pair. Thus, there are 4 complete pairs of socks not missing. Now, the number of ways of selecting 2 socks from a total of 5 x 2 = 10 socks is given by 10C2 = 45 The number of ways of selecting 2 socks from 10 sucks such that the two socks are of the same pair is the same as the probability of selecting 1 pair of socks from 5 pairs and is given by 5C1 = 5 Therefore, the probability of the best case scenario is given by 5 / 45 = 1 / 9

**Part B: The probability on the worst case scenario**

The worst case scenario results when the two missing socks are not of the same pair. Thus, there are 3 complete pairs of socks not missing. Now, the number of ways of selecting 2 socks from a total of 5 x 2 = 10 socks is given by 10C2 = 45 The number of ways of selecting 2 socks from 10 sucks such that the two socks are not of the same pair is given by 10C2 - 5C1 = 45 - 5 = 40 Therefore, the probability of the worst case scenario is given by 40 / 45 = 8 / 9

16. Give an example of an algorithm that should not be considered an application of the brute-force approach.

Euclid’s algorithm and the standard algorithm for finding the binary representation of an integer are examples from the algorithms previously mentioned in this book. There are, of course, many more examples in its other chapters.

17. Sort the list [22, 3, 4, -7, 90, 5, 0, 11] using (i) bubble sort, (ii) merge sort.

18. How many comparisons (both successful and unsuccessful) will be made by the brute-force algorithm in searching for each of the following patterns in the binary text of one thousand zeros?

a) 00001 The Search String (0000….000) is of length 1000; the Search Pattern (00001) is of length 5. Hence, there will be 1000-5+1 = 996 iterations. In each of these iterations, the first 4 comparisons would be successful and the last comparison will be unsuccessful. Hence, there will be 996\*4 = 3,984 successful comparisons and 996\*1 = 996 unsuccessful comparisons. Total comparisons = 3984 + 96 = 4980 comparisons.

b) 10000 There will be a total of 1000-5+1 = 996 iterations. In each of these iterations, the first comparison would itself be unsuccessful. Hence, there will be 996\*1 = 996 unsuccessful comparisons and there will not be any successful comparisons. Total comparisons = 996.

c) 01010 There will be a total of 1000-5+1 = 996 iterations. In each of these iterations, the first comparison would be successful and the second comparison would be unsuccessful. Hence, there will be 996\*1 = 996 successful comparisons and another 996\*1 = 996 unsuccessful comparisons. Total comparisons = 1992.

19. What are some real-world applications of closest-pair problem?

The closest-pair solution has many applications in real-life. It forms a main step in many

problem-solving procedures. These include applications in air/land/water traffic-control systems.

A traffic control system can use the solution in order to avoid collisions between vehicles. The

algorithm has applications in detecting collisions after they happen. There are also applications in

self-navigating vehicles. The solution also has applications in bodies which must always keep

close to particular other bodies. The problem also has applications in imaging technologies,

pattern recognition, CAD, VLSI.

20. What is knapsack problem definition?

The **knapsack problem** or rucksack **problem** is a **problem** in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.