

# University of Sargodha

## BS 4<sup>th</sup> Term Examination 2016

Subject: Computer Science

Paper: Design & Analysis of Algorithm (CS: 3143)

\*\*\*\*\*

### **Q1. Write short answers of the following.**

#### **A. Name any 3 algorithms?**

There are some following algorithms:

1. Kruskal's Algorithm
2. Prim's Algorithm
3. Dijkstra's Algorithm

#### **B. What is a heuristic?**

Heuristic enables a person to discover or learn something for themselves. A heuristic algorithm is one that is designed to solve a problem in a faster and more efficient fashion than traditional methods by sacrificing optimality, accuracy, precision, or completeness for speed.

#### **C. Name any linear data structure.**

A data structure is said to be linear if its elements form a sequence or a linear list. Examples:

- ✓ Array
- ✓ Linked List
- ✓ Stacks
- ✓ Queues

#### **D. Name any two heuristics.**

There are the following heuristics:

- ✓ Virus Scanning
- ✓ Tabu Search
- ✓ Swarm Intelligence
- ✓ Artificial Neural Networks

#### **E. What are the conditions when we use dynamic programming instead of greedy algorithms?**

We use dynamic programming instead of greedy algorithms to solve a problem where it is applicable to problems with the property that:

- ✓ It can be partitioned into sub-problems (probably in more than one way),
- ✓ Those sub-problems can be solved independently,
- ✓ (optimal) solutions of those sub-problems can be combined to (optimal) solutions of the original problem and
- ✓ Sub-problems have the same property (or are trivial).

#### **F. What is Big O notation?**

Big O Notation is the language we use to describe the complexity of an algorithm. In other words, Big O Notation is the language we use for talking about how long an algorithm takes to run. It is how we compare the efficiency of different approaches to a problem. With Big O Notation we express the runtime in terms of how quickly it grows relative to the input, as the input gets larger.

#### **G. What is Non Deterministic Computer?**

Non deterministic computer is the one which works of non-deterministic algorithms. A nondeterministic algorithm can provide different outputs for the same input on different executions. Nondeterministic algorithms are useful for finding approximate solutions, when an exact solution is difficult or expensive to derive using a

*Solved by Talha Shahab*

deterministic algorithm. A nondeterministic algorithm is capable of execution on a deterministic computer which has an unlimited number of parallel processors.

#### **H. What is average case efficiency?**

The average-case complexity of an algorithm is the amount of some computational resource (typically time) used by the algorithm, averaged over all possible inputs. It is frequently contrasted with worst-case complexity which considers the maximal complexity of the algorithm over all possible inputs. The *average-case complexity* of the algorithm is the function defined by the average number of steps taken on any instance of size  $n$ .

#### **I. What are NP problems?**

A problem is assigned to the NP (nondeterministic polynomial time) class if it is solvable in polynomial time by a nondeterministic Turing machine. NP is set of decision problems that can be solved by a Non-deterministic Turing Machine in Polynomial time. Informally, NP is set of decision problems which can be solved by a polynomial time via a "Lucky Algorithm", a magical algorithm that always makes a right guess among the given set of choices.

#### **J. Is space complexity affected by size of cache memory?**

No, space complexity is not affected by size of cache memory. Space complexity is the space required to store instructions and data to execute the program. It is affected by number of instructions and data to execute a program.

#### **K. Is space complexity affected by size of registers?**

No, space complexity is not affected by size of registers. Space complexity is the space required to store instructions and data to execute the program. It is affected by number of instructions and data to execute a program.

#### **L. Name any divide and conquer technique.**

There are some techniques based upon divide and conquer:

- ✓ Binary Search
- ✓ Quick Sort
- ✓ Merge Sort

#### **M. Give an example where recursion is used.**

We can use recursion to find factorials, binary search, merge sort and to find greatest common divisor.

#### **N. What is good solution?**

The TRIZ principles list the common characteristics of a good solution as:

- ✓ Resolve a contradiction
- ✓ Increase the "ideality" of the system
- ✓ Uses idle, easily available resources

#### **O. Discuss any two properties of an algorithm.**

In order for an algorithm to be useful, it must help us find a solution to a specific problem. For that to happen, an algorithm must satisfy five properties.

- ✓ **Input:** The inputs used in an algorithm must come from a specified set of elements, where the amount and type of inputs are specified.
- ✓ **Output:** The algorithm must specify the output and how it is related to the input.
- ✓ **Definiteness:** The steps in the algorithm must be clearly defined and detailed.
- ✓ **Effectiveness:** The steps in the algorithm must be doable and effective.
- ✓ **Finiteness:** The algorithm must come to an end after a specific number of steps.

#### **P. What is radix sort?**

The idea of Radix Sort is to do digit by digit sort starting from least significant digit to most significant digit. Radix sort uses counting sort as a subroutine to sort. Radix sort is an integer sorting algorithm that sorts data with

*Solved by Talha Shahab*

integer keys by grouping the keys by individual digits that share the same significant position and value (place value).

### **Subjective Part**

2. Explain how dynamic programming can get the solution of  $10!$  Quickly as compared to greedy algorithm.
3. Show Huffman coding for a data string that has following character frequency. Use greedy technique.

Character	Frequency
A	24
B	12
C	10
D	8
E	8

4. Explain Bellman-Ford algorithm with help of an example.
5. Discuss substitution method for solving recurrences with help of an example.
6. Explain Rabin-Karp algorithm of string matching with help of an example.



Visit

**<https://tshahab.blogspot.com>**

for more.