Past papers A-O-A Aoa paper lecture note for students

Design Analysis of Algorithm (University of Sargodha)

Scan to open on Studocu

Subject:

Design and Analysis Of Algorithm

Past Papers :

2015

2016

2017

2018

---***---***---***----****---------

2015 Past Papers

1)- write down steps of designing of an algorithm?

1. **Design**. The first stage is to identify the problem and thoroughly understand it. ...
2. **Analyze**. Once you have the basic framework of the **algorithm** it's time to start analyzing how efficient the code is in solving the problem. ...
3. **Implement**. Writing and coding the **algorithm** is the next **step** in the process. ...
4. **Experiment.**

2)- Define standard notation?

Standard notation is when a number is completely written out using numerical digits. Some examples of numbers written in **standard notation** are 64,100 and 2,000,000. **Standard notation** is commonly used in everyday math. However, when working with large numbers, it can become cumbersome to write out every single digit.

3)- Define role of algorithm in computing?

1. To improve the efficiency of a computer program

Another way of looking at the efficiency of software is speed. An algorithm can be used to improve the speed at which a program executes a problem. A single algorithm has the potential of reducing the time that a program takes to solve a problem.

2.Proper utilization of resources

The right choice of an algorithm will ensure that a program consumes the least amount of memory. Apart from memory, and the algorithm can determine the amount of processing power that is needed by a program.

4) why we use substitution methods for solving recurennences?

The substitution method for solving recurrences is famously described using two steps:

1. Guess the form of the solution.
2. Use induction to show that the guess is valid.

This method is especially powerful when we encounter recurrences that are non-trivial and unreadable via the master theorem. We can use the substitution method to establish both upper and lower bounds on recurrences.-

5) What is Best and Worst case complexity of quick sort algo?

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

6) big O and big Omega on notes…

7) What is Dynamic programming?

Dynamic Programming refers to a very large class of algorithms. The idea is to break a large

problem down (if possible) into incremental steps so that, at any given stage, optimal solutions are known to *sub-problems*. When the technique is applicable, this condition can be extended incrementally without having to alter previously computed optimal solutions to subproblems. Eventually the condition applies to all of the data and, if the formulation is correct, this together with the fact that nothing remains untreated gives the desired answer to the complete problem.

8)- Define minimum spanning tree?

Given an undirected and connected graph G=(V,E), a spanning tree of the graph G is a tree that spans G (that is, it includes every vertex of G) and is a subgraph of G (every edge in the tree belongs to G). The cost of the spanning tree is the sum of the weights of all the edges in the tree. There can be many spanning trees. Minimum spanning tree is the spanning tree where the cost is minimum among all the spanning trees. There also can be many minimum spanning trees.Minimum spanning tree has direct application in the design of networks

9)-Define sparse Graph?

A *graph* in which the number of *edges* is much less than the possible number of edges. *A directed graph can have at most n(n-1) edges, where n is the number of vertices. An undirected graph can have at most n(n-1)/2 edges.*

Graph: A collection of vertices and edges

- Vertex: A simple object that can have a name and other properties
- Edge: A connection between two vertices
- Path: A list of vertices in which successive vertices are connected by edges in the graph
- Connected Graph: A graph in which there is a path from every node to every other node in the graph.
- Connected Component: A set of vertices within a graph in which:
    1. there is a path from every vertex within the set to every other vertex in the set, and
    2. there is not a path from any vertex within the set to any vertex not in the set

  If a graph is not connected, then it has at least two connected components.

- Cycle:- A path in which the first and last vertices are the same (i.e., a path from a vertex back to itself)
- Undirected Graph: A graph in which the edges are undirected (i.e., bidirectional).
- Directed Graph: A graph in which the edges are directed (i.e., unidirectional). If edges are directed, then we speak of the edge as going *from* one vertex *to* another vertex.
- Dense Graph: Roughly speaking, a graph in which the number of edges is greater than or equal to V lg V, where V is the number of vertices.
- Sparse Graph: Roughly speaking, a graph in which the number of edges is < V lg V, where V is the number of vertices
- .----------2015----------3

10) What is topological sort?

Topological sorting for Directed Acyclic Graph (DAG) is a linear ordering of vertices such that for every directed edge uv, vertex u comes before v in the ordering. Topological Sorting for a graph is not possible if the graph is not a DAG.

For example, a topological sorting of the following graph is “5 4 2 3 1 0”. There can be more than one topological sorting for a graph. For example, another topological sorting of the following graph is “4 5 2 3 1 0”.

11) What is Binary Tree?

A **binary search**, also known as a **half-interval search**, is an algorithm used in computer science to locate a specified value (key) within an array. For the search to be binary, the array must be sorted in either ascending or descending order.

12)- Diff. b/w Bucket sort and Radix sort?

The initial pass of both `RadixSort` and`BucketSort` is exactly the same. The elements are put in `buckets` (or `bins`) of incremental ranges (e.g. 0-10, 11-20, ... 90-100), depending on the number of digits in the largest number.
In the next pass, however, `BucketSort`orders up these 'buckets' and appends them into one array. However, `RadixSort` appends the buckets without further sorting and 're-buckets' it based on the second digit (ten's place) of the numbers. Hence, BucketSort is more efficient for 'Dense' arrays, while RadixSort can handle sparse (well, not exactly sparse, but spaced-out) arrays well.
Bucket Sort and Radix Sort are like sister sorting algorithms because they are not comparison sorts and the general idea is similar. Also, they both are a bit abstract in implementation.

13) Name Three Elementary Graph Object

Ans:-

14)Draw directed cyclic graph

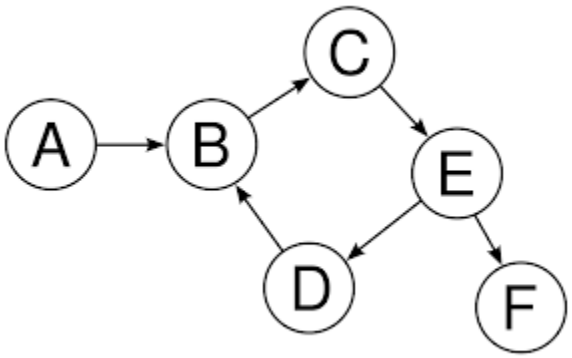


Figure 3 : Directed Graphs

15) Write elements of Greedy strategy?

- Optimal Substructure: An optimal solution to the problem contains within it optimal solutions to sub-problems. ...
- The 0 - 1 knapsack problem: A thief has a knapsack that holds at most W pounds. ...
- Fractional knapsack problem: takes parts, as well as wholes.

16) What is heap and heap sort?

A sorting algorithm that works by first organizing the data to be sorted into a special type of binary tree called a *heap*. The heap itself has, by definition, the largest value at the top of the tree, so the heap sort algorithm must also reverse the order. It does this with the following steps:

1. Remove the topmost item (the largest) and replace it with the rightmost leaf. The topmost item is stored in an array.
2. Re-establish the heap.
3. Repeat steps 1 and 2 until there are no more items left in the heap.

The sorted elements are now stored in an array.

A heap sort is especially efficient for data that is already stored in a binary tree. In most cases, however, the *quick sort* algorithm is more efficient.

----------2015----------5

2016 past papers

1)-Name any 3 algo?

- Simple recursive algorithms.
- Backtracking algorithms.
- Divide and conquer algorithms.
- Dynamic programming algorithms.
- Greedy algorithms.
- Branch and bound algorithms.
- Brute force algorithms.
- Randomized algorithms.

2)-What is s heuristic?

Heuristics are a problem-solving method that uses shortcuts to produce good-enough solutions given a limited time frame or deadline. Heuristics are a flexibility technique for quick decisions, particularly when working with complex data. Decisions made using an heuristic approach may not necessarily be optimal. Heuristic is derived from the Greek word meaning “to discover”.

3) name any linear data structure?

- Arrays.
- Queues.
- Stacks.
- Linked lists.

4)- name any two heuristic?

Types of Heuristics: Availability Heuristics , Representativeness Heuristics & Base-Rate Heuristics

5) **What are the condition when we use dynamic programming instead of greedy algo?**

1. Greedy algorithm is one which finds feasible solution at every stage with the hope of finding optimal solution whereas Dynamic programming is one which break the problems into series of overlapping sub-problems

----------2016----------1

2. Greedy algorithm never reconsiders its choices whereas Dynamic programming may consider the previous state.
3. Greedy algorithm is less efficient whereas Dynamic programming is more efficient.
4. Greedy algorithm have a local choice of the sub-problems whereas Dynamic programming would solve the all sub-problems and then select one that would lead to an optimal solution.
5. Greedy algorithm take decision in one time whereas Dynamic programming take decision at every stage.
6. Greedy algorithm work based on choice property whereas Dynamic programming work based on principle of optimality.
7. Greedy algorithm follows the top-down strategy whereas Dynamic programming follows the bottom-up strategy.

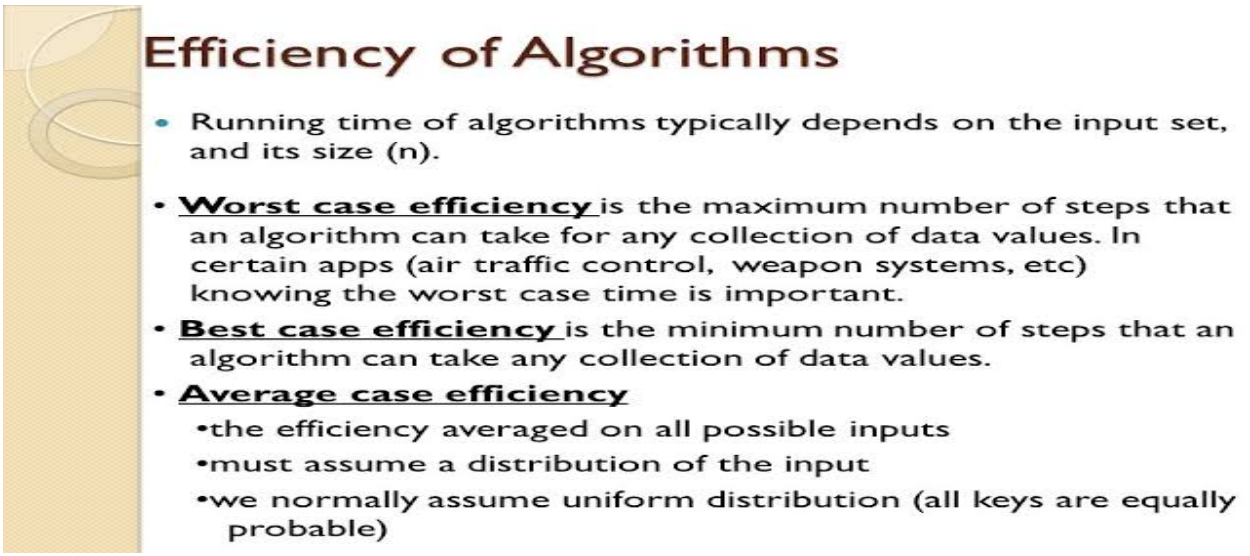
6) what is Big-o-Notions?

Big O notation is a mathematical **notation** that describes the limiting behavior of a function when the argument tends towards a particular value or infinity.

7) What is Non deterministic computer?

In computer science, a **nondeterministic algorithm** is an algorithm that, even for the same input, can exhibit different behaviors on different runs, as opposed to a deterministic algorithm. There are several ways an algorithm may behave differently from run to run. A concurrent algorithm can perform differently on different runs due to a race condition. A probabilistic algorithm's behaviors depends on a random number generator. An algorithm that solves a problem in nondeterministic polynomial time can run in polynomial time or exponential time depending on the choices it makes during execution. The nondeterministic algorithms are often used to find an approximation to a solution, when the exact solution would be too costly to obtain using a deterministic one.

8)- What is average case efficiency?



9)- What is NP problem?

A problem is assigned to the NP (nondeterministic polynomial time) class if it is solvable in polynomial time by a nondeterministic Turing machine.

A P-problem (whose solution time is bounded by a polynomial) is always also NP. If a problem is known to be NP, and a solution to the problem is somehow known, then demonstrating the correctness of the solution can always be reduced to a single P (polynomial time) verification. If P and NP are *not* equivalent, then the solution of NP-problems requires (in the worst case) an exhaustive search.

10) Is space complexity affected by size of cache memory?

Memory efficiency and locality have substantial impact on the performance of programs, particularly when operating on large data sets. The widespread deployment of multi-core machines, however, brings new challenges. Specifically, since the memory (RAM) is shared across multiple processes, the effective memory-size allocated to each process fluctuates over time.Memory fluctuations are the norm on most computer systems. Each process's share of memory changes dynamically as other processes start, stop, or change their own demands for memory

11) Is space complexity affected by size of register?

Ans:-_____

12) Name any Divide and Conquer techniques?

1) Binary Search 2) Quicksort 3) Merge Sort 4) Strassen's Algorithm

13)- Give an example where recursion is used?

Recursion is a common method of simplifying a problem into subproblems of same type. This is called divide and conquer technique. A basic example of recursion is factorial function.

Example:

```
int factorial(int n){

if(n==0) return 1;

else return (n* factorial(n-1));
```

14)What is Good Solution?

Heuristics are the weapon of choice when it comes to solving complex combinatorial optimization problems. With knowledge we hereby mean that we can defi?ne desirable structural characteristics of good solutions. Our knowledge generation approach is based on data mining and we demonstrate its concept with the help of the most prominent combinatorial problem in Operations Research, the Vehicle Routing Problem. We de?fine metrics to measure a solution and an instance

15)Discuss any two properties of an algorithm?

- Finiteness. An **algorithm** must always terminate after **a** finite number of steps.
- Definiteness. Each step of an **algorithm** must be precisely defined; **the** actions to be carried out must be rigorously and unambiguously specified for each case.
- Input. ...
- Output. ...
- Effectiveness.

16) what is Radix sort?

In computer science, **radix sort** is a non-comparative sorting algorithm. It avoids comparison by creating and distributing elements into buckets according to their radix. For elements with more than one significant digit, this bucketing process is repeated for each digit, while preserving the ordering of the prior step, until all digits have been considered. For this reason, **radix sort** has also been called **bucket sort** and **digital sort**.

2017 Past Papers

1)Big-o-notions ?

Big O notation is a mathematical **notation** that describes the limiting behavior of a function when the argument tends towards a particular value or infinity.

2) **What is meant by problems of optimality**

- **Definition: A problem is said to satisfy the Principle of Optimality if the subsolutions of an optimal solution of the problem are themesleves optimal solutions for their subproblems.**
- **Examples:**
  - **The shortest path problem satisfies the Principle of Optimality.**

3) **What r the diff. criteria used to improve the complexity of algo?**

An algorithm must satisfy the following criteria:

1. Input- These are the values that are supplied externally to the algorithm.

2. Output- These are the results that are produced by the algorithm.

3. Definiteness- Each step must be clear and unambiguous.

4. Finiteness- The algorithm must terminate after a finite number of steps.

5. Effectiveness- Each step must be feasible i.e. it should be practically possible to perform the step.

4)-write down the recursive solution for knapsack problem?

1. Let, fi (yj)be the value of optimal solution.
2. Using formula: $f_i(y_j)=\max f_{i-1}(y), f_{i-1}(y-w_i)+p_i f_i(yj)=\max fi-1(y), fi-1(y-wi)$ +pi to solve problem.
3. Then Si is a pair (p, w) where $p=f(y_j)$p=f(yj) and $w=y_j$w=yj
4. Initially $S_0=(0,0)$S0=(0,0)
5. Then $S_{i1}=(P,W)|(P-p_i,W-w_i)S_iS1i=(P,W)|(P–pi,W–wi)Si$
6. S_{i+11}S1i+1 can be computed by merging S_i and S_{i1}Si and S1i
7. This is used for obtaining optimal solution.

5)What is the time comnplexity of prims algo?

Time complexity of Prim's Algorithm

Running Time = O(E + V log V) (E = #edges, V = #nodes)

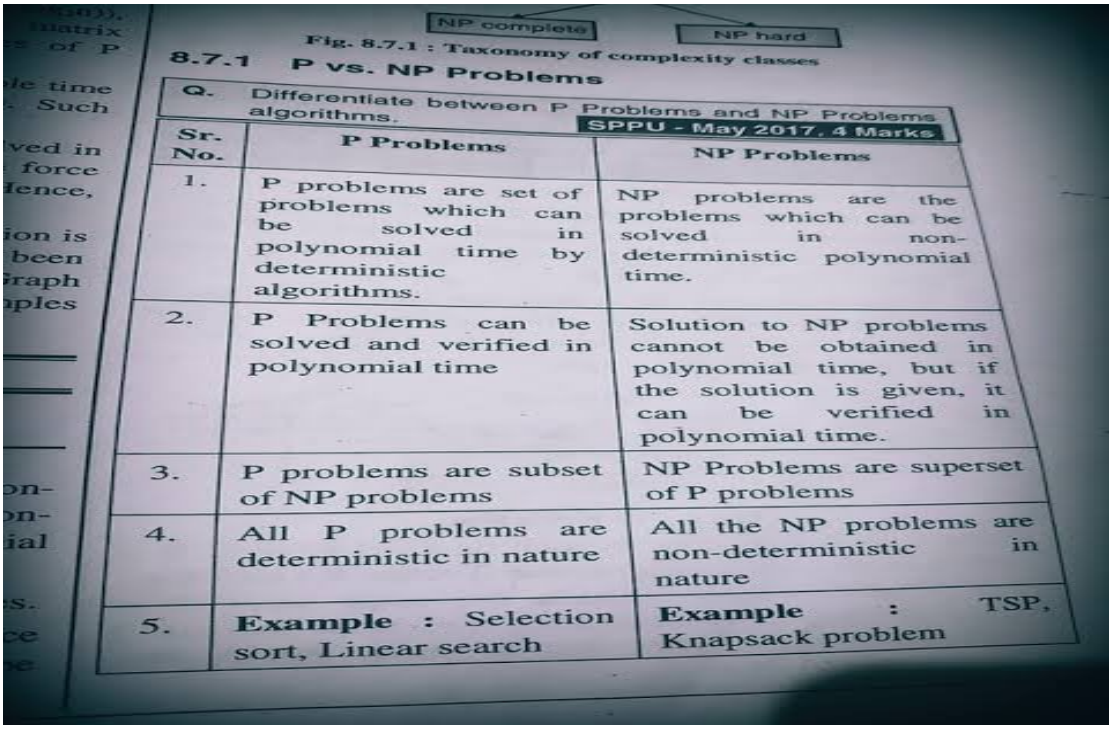
Minimum edge weight data structure	Time complexity (total)
adjacency matrix, searching	$O(\|V\|^2)$
binary heap and adjacency list	$O((\|V\| + \|E\|) \log \|V\|) = O(\|E\| \log \|V\|)$
Fibonacci heap and adjacency list	$O(\|E\| + \|V\| \log \|V\|)$

6) Write down the diff. b/w the greedy method and dynamic programming?

DYNAMIC PROGRAMMING VS GREEDY

Dynamic Programming	Greedy Algorithm
At each step, the choice is determined based on solutions of sub problems.	At each step, we quickly make a choice that currently looks best. A local optimal (greedy) choice
Sub-problems are solved first.	Greedy choice can be made first before solving further sub-problems.
Bottom-up approach	Top-down approach
Can be slower, more complex	Usually faster, simpler

7) **Diff. b/w P and NP problem**



NP complete NP hard

Fig. 8.7.1 : Taxonomy of complexity classes

8.7.1 P vs. NP Problems

Q. Differentiate between P Problems and NP Problems algorithms. **SPPU - May 2017, 4 Marks**

Sr. No.	P Problems	NP Problems
1.	P problems are set of problems which can be solved in polynomial time by deterministic algorithms.	NP problems are the problems which can be solved in non-deterministic polynomial time.
2.	P Problems can be solved and verified in polynomial time	Solution to NP problems cannot be obtained in polynomial time, but if the solution is given, it can be verified in polynomial time.
3.	P problems are subset of NP problems	NP Problems are superset of P problems
4.	All P problems are deterministic in nature	All the NP problems are non-deterministic in nature
5.	**Example** : Selection sort, Linear search	**Example** : TSP, Knapsack problem

8)- What is meant by Minimum spanning tree?

A **minimum spanning tree** (MST) or **minimum** weight **spanning tree** is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the **minimum** possible total edge weight.

9)-How problem are solved using Divide and Conquer approach?

A **divide-and-conquer algorithm** works by recursively breaking down a **problem** into two or more sub-**problems** of the same or related type, until these become simple enough to be **solved** directly. The solutions to the sub-**problems** are then combined to give a solution to the original **problem**.

10)- Define all pair shorted path problem?

The all-pairs shortest path problem is the determination of the shortest ***graph distances*** between every pair of vertices in a given graph. The problem can be solved using n applications of ***Dijkstra's algorithm*** or all at once using ***the Floyd-Warshall algorithm***. The latter algorithm also works in the case of a weighted graph where the edges have negative weights.

The matrix of all distances between pairs of vertices is called the ***graph distance matrix*****,** or sometimes the all-pairs shortest path matrix.

11)-What is chained matrix multiplication?

Matrix chain multiplication (or **Matrix Chain** Ordering Problem, MCOP) is an optimization problem that can be solved using dynamic programming. Given a sequence of **matrices**, the goal is to find the most efficient way to **multiply** these **matrices**. ... There are many options because **matrix multiplication** is associative.

12)What is the purpose of Dijkstra Algo?

Dijkstra's algorithm can be used to determine the shortest path from one node in a graph to every other node within the same graph data structure, provided that the nodes are reachable from the starting node. **Dijkstra's algorithm** can be used to find the shortest path.

13)- Write an algo. Using recursive function to find sum of n number?

```
set k = n
set sum = 0
      while k > 0
            set sum = sum + k
            set k = k -1
      end while loop
return sum
```

14)- Write down the ingredients of Dynamic Programming?

Two key ingredients that an optimization problem must have in order for dyn amic programming to apply: optimal substructure and overlapping sub proble ms.

15) What is activity selection problem?

The ***activity selection problem*** is a combinatorial optimization ***problem*** concerning the ***selection*** of non-conflicting ***activities*** to perform within a given time frame, given a set of ***activities*** each marked by a start time (s_i) and finish time (f_i).

16)- Write the Recurrence relation of Quick sort?

2018 past paper

1:- What is algorithm optimality?

An algorithm for solving a problem P is *asymptotically optimal with respect to worst-case running time* if it runs in time $T(n)$, and any algorithm for P runs in time $\Omega(T(n))$.

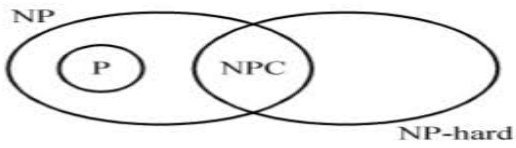
2) What are the different complexities according to which we analyse any algorithm?

Ans:- worst-case complexity big oh, best-case complexity big omega, and average-case complexity big theta.

3) what is merge sort? And insertion sort is better than merge sort?

Mergesort is a stable, out-of-place **sorting** algorithm with a time complexity of O(N log N) and an extra space complexity of O(N). If n is large, memory is not a constraint and the array completely unordered, MergeSort is preferred over InsertionSort.

4)What is NP hard and NP complex problem?



- P: the class of problems which can be solved by a deterministic polynomial algorithm.
- NP : the class of decision problem which can be solved by a non-deterministic polynomial algorithm.
- NP-hard: the class of problems to which every NP problem reduces.
- NP-complete (NPC): the class of problems which are NP-hard and belong to NP.

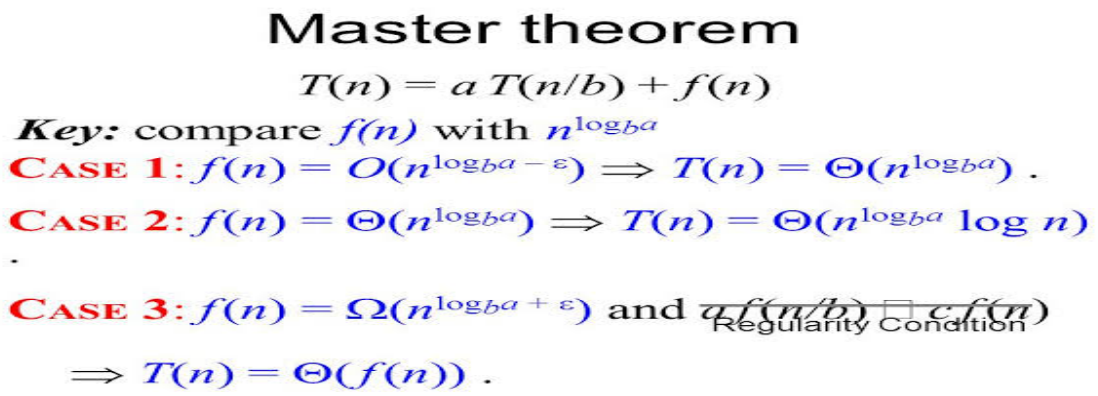
8- 2

5) Define feasible and optimal solution?

A **solution** (set of values for the decision variables) for which all of the constraints in the Solver model are satisfied is called a **feasible solution**. ... An **optimal solution** is a **feasible solution** where the objective function reaches its maximum (or minimum) value – for example, the most profit or the least cost.

---------- 1 ----------

6) Write down three cases of master method for solving recurrences?



7) which sorting algorithm will perform better if array is already sorted?

The **best** case input is an array **that** is **already sorted**. In this case insertion **sort** has a linear running time (i.e., O(n)).

8) what are the drawback of dynamic programming?

It takes a lot of memory to store the calculated result of every subproblem without ensuring if the stored value will be utilized or not.

- Many times, output value gets stored and never gets utilized in the next subproblems while execution. It leads to unnecessary memory utilization.
- In DP, functions are called recursively. Stack memory keeps increasing.

9) Different b/w dynamic programming and greedy approach?

DYNAMIC PROGRAMMING VS GREEDY

Dynamic Programming	Greedy Algorithm
At each step, the choice is determined based on solutions of sub problems.	At each step, we quickly make a choice that currently looks best. A local optimal (greedy) choice
Sub-problems are solved first.	Greedy choice can be made first before solving further sub-problems.
Bottom-up approach	Top-down approach
Can be slower, more complex	Usually faster, simpler

---------- 2----------(2018)

10) What is the time complexity of prims algo?

Time complexity of Prim's Algorithm

Running Time = O(E + V log V) (E = #edges, V = #nodes)

Minimum edge weight data structure	Time complexity (total)
adjacency matrix, searching	$O(\|V\|^2)$
binary heap and adjacency list	$O((\|V\| + \|E\|) \log \|V\|) = O(\|E\| \log \|V\|)$
Fibonacci heap and adjacency list	$O(\|E\| + \|V\| \log \|V\|)$

11) How problem are solving using Divide and Conquer approach?

A **divide-and-conquer algorithm** works by recursively breaking down a **problem** into two or more sub-**problems** of the same or related type, until these become simple enough to be **solved** directly. The solutions to the sub-**problems** are then combined to give a solution to the original **problem**.

12) What are the constantans of knapsack problem?

If there is more than one constraint (for example, both a volume limit and a weight limit, where the volume and weight of each item are not related), we get the **multiply-constrained knapsack**

problem, **multidimensional knapsack problem**, or *m*-**dimensional knapsack problem**. (Note, "dimension" here does not refer to the shape of any items.) This has 0-1, bounded, and unbounded variants

13:-Define all pair shorted path problem?

The all-pairs shortest path problem is the determination of the shortest graph distances between every pair of vertices in a given graph. The problem can be solved using n applications of Dijkstra's algorithm or all at once using the Floyd-Warshall algorithm. The latter algorithm also works in the case of a weighted graph where the edges have negative weights.

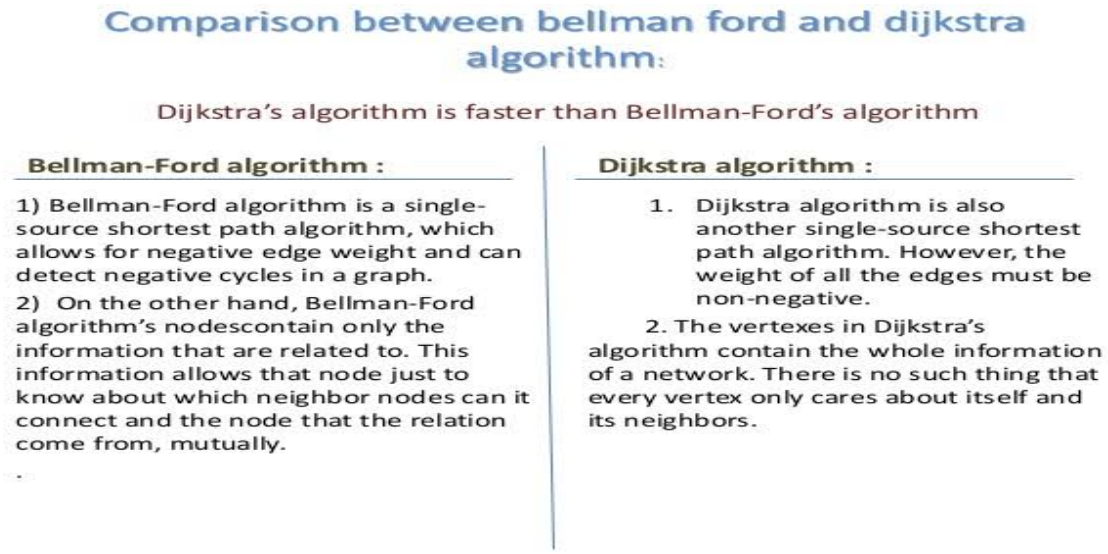
The matrix of all distances between pairs of vertices is called the graph distance matrix, or sometimes the all-pairs shortest path matrix.

14) Write down an algo using recursive function to find sum of n number?

- ---------- 3 (2018)---------- ----

```
set k = n
set sum = 0
    while k > 0
        set sum = sum + k
        set k = k -1
    end while loop
return sum
```

15) What is the major different b/w Dijkstra and Bellman ford?



16) write down the recursive solution for Floyd warshall algo?

Ans:-

1. n = W.rows
2. $D^{(0)}$=W
3. For k = 1 to n
4. let $D^{(k)} = (d_{ij}^{(k)})$ be a new n x n matrix
5. For i = 1 to n
6. For j = 1 to n
7. $d_{ij}^{(K)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$
8. return $D^{(n)}$

---------- 4 (2018)----------