Name: Ayush Jain
SAP ID: 60009200132
Div: B
Computer Engineering

AA - Assignment 2

Q.1) Discuss the technique to find the closest pair of points?

→ • The Euclidian b/w two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ is $d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

• A brute force approach is to compute the distance b/w two random points and take minimum. However, running time is $\Theta(n^2) = O(n^2)$

• A high level description of a much better algorithm is given below:

• let Q be a set of planar points. $|Q| \leq 3$, then the distances b/w all pairs of points are computed and the closest pair is reported. If $|Q| > 3$, we use divide and conquer and continue the procedure.

• Each recursive call receives as i/p.

• Set $P \subseteq Q$

• Array X and Y containing points P sorted by x and y co-ordinates respectively.

Q.2) Explain Travelling Salesperson problem as an approximation problem.

→ i) Travelling Salesperson problem is a graph computational problem where the salesman needs to visit all cities in a list exactly once and the distances between all the cities are known. The goal is to find the shortest possible route in which the salesman visits all the

cities and return to the initial city.

2) The approximate sol. to the problem works only if the problem instances satisfy triangle inequality.

3) dist(i,j) ≤ dist(i,k) + dist(k1,k2) + ... dist(k,j)

When the cost function satisfies the triangle inequality, we can design an approx algo that returns a path whose cost is never more than twice the cost of an optimal flow.

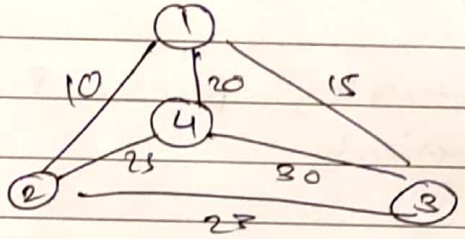
4) The algorithm is:

(1) Let i be the starting and ending point for salesman.

(2) construct MST from graph with 1 as root using prim's algo.

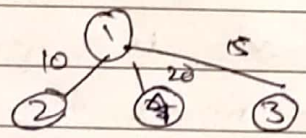
(3) list vertices visited in pre-order walk of the constructed MST and add 1 at the end.

Let us consider the foll. example:



pre-order traversal is
1 → 2 → 4 → 3 → 1

Minimum spanning tree:



The approximation algo may not always provide the optimal tour.

Q. 3) Write a short note on:

(1) Competitive Ratio:

→ • In the context of algorithm, the competitive ratio is a measure of the performance of an online algorithm compared to an optimal algorithm. Online algorithm are designed to work with incomplete information making decisions on the fly as new data becomes available.
• The ratio is designed as the worst case ratio of the cost of online algorithm (A) to the cost of the optimal offline algorithm.

Competitive ratio = max(A / OPT)

• The ratio is always greater than or equal to 1, as the running cost of A is greater or equal to the optimal cost.
• The ratio is a useful tool for comparing the performance of different online algorithms to the same problem. It can also provide insight into the difficulty of a problem and inherent limitations of online algorithms.

(2) K - Server:

→ • The K-Server problem is a classical problem in theoretical computer science that deals with the online management of K-server or a mutable space. The problem can be stated as:

• Given a set of n points in a metric space and k servers, the goal is to service a sequence of requests for the points, such that each request is received by one of the k servers, and the cost of servicing a request is proportional to the distance between the requested point and the server that is used to service it.

• The problem is an np-hard problem which means that there is no known algo that can solve the problem in polynomial time, unless P = NP.

• One of the widely used algo is the greedy problem which works as follows:

→ Given a request for a point P, the algo choises the server that is closest to P, and mooves the server to P.

→ If there are ties, the algo arbitiary choose the closest servers.

→ The greedy approach has a competitive ratio of k, which means that produced by the algo is at most k times a optimal cost.

→ Not the best but a good sol. considering the problem has. The k-server has important applications in computer network, distributed computing and logistics.