

Questions:

1. Order the following functions in the increasing order of growth (no justification needed): **6 marks**

$$n^2, 2^{\log^2 n}, n^{1/3}, n^{\log \log n}$$

2. Solve the following recurrence by recursion tree method, and verify your answer using Master method: **6 marks**

$$T(n) = 5T\left(\frac{n}{7}\right) + n^2$$

3. You need to find k largest elements from an unsorted array of n distinct elements. Design an efficient algorithm for the same. There is no credit for a suboptimal algorithm. **8 marks**

4. You are given n distinct points and one line ℓ on the plane, and some constant $r > 0$. Each of the n points is within (perpendicular) distance r of the line. You need to place disks of radius r with their centers on line ℓ such that each of the n points lies within at least one disk.

Design an $O(n \log n)$ time greedy algorithm to find the minimum number of disks required. Prove its optimality. **10 marks**

Questions:

1. Order the following functions in the increasing order of growth (no justification needed): **6 marks**

$$n^2, 2^{\log^2 n}, n^{1/3}, n^{\log \log n}$$

2. Solve the following recurrence by recursion tree method, and verify your answer using Master method: **6 marks**

$$T(n) = 5T\left(\frac{n}{7}\right) + n^2$$

3. You need to find k largest elements from an unsorted array of n distinct elements. Design an efficient algorithm for the same. There is no credit for a suboptimal algorithm. **8 marks**

4. You are given n distinct points and one line ℓ on the plane, and some constant $r > 0$. Each of the n points is within (perpendicular) distance r of the line. You need to place disks of radius r with their centers on line ℓ such that each of the n points lies within at least one disk.

Design an $O(n \log n)$ time greedy algorithm to find the minimum number of disks required. Prove its optimality. **10 marks**