COL351: Analysis and Design of Algorithms

Tutorial Sheet - 1

August 7, 2022

Question 1 Consider a modified implementation of the Merge Sort where the splits of an input array of size n are sub-arrays of size $\Theta(\sqrt{n})$, $\Theta(n-\sqrt{n})$. Provide an $O(n\sqrt{n})$ bound on the running time of the algorithm.

```
1  Set n = length(A) and K = ⌊√n⌋;
2  if n ≤ 5 then Sort the array in Brute-force manner and return;
3  Store in B_1 the sub-array A[ 0, K];
4  Store in B_2 the sub-array A[ K + 1, n − 1 ];
5  MergeSort(B_1);
6  MergeSort(B_2);
7  Set x, y, pos = 0;
8  while (x < length(B_1) or y < length(B_2)) do
9      if (B_1[x] ⩽ B_2[y] and x < length(B_1)) then
10         set A[pos] = B_1[x], and increment pos and x by 1;
11     else set A[pos] = B_2[y], and increment pos and y by 1;
12 end
```

Algorithm 1: MergeSort(A)

Hint: Show that the recurrence relation $H(n) = 1 + H(n - \lceil\sqrt{n}\rceil)$ satisfies : $H(n) \leq 2\sqrt{n}$.

Question 2 The manager of a student union in IIT Delhi is in-charge of a group of n students, each of whom is scheduled to work one shift during the week. Each shift is a single contiguous interval of time. There can be multiple shifts going on at once. The Manager is trying to choose a *minimum* subset of these n students to form a supervising committee that she can meet weekly. She considers such a committee to be *complete* if, shift of each student not in the committee overlaps (at least partially) with the shift of some student who is in the committee. In this way, each student's performance can be observed by at least one person who's serving in the committee.

(a) Formulate this problem mathematically.

(b) Give an efficient algorithm that takes the schedule of n shifts and produces a complete supervising committee containing as few students as possible.

Question 3 Suppose there are n jobs $J_1 = [s_1, t_1], \ldots, J_n = [s, t_n]$ satisfying $t_1 \leqslant \cdots \leqslant t_n$. Design an $O(n)$ time algorithm to compute an optimal scheduling for jobs $(J_1, \ldots, J_n)$.

Question 4 A "dominating set" for an undirected graph $G = (V, E)$ is a set S of vertices such that for each $v \in V$, either v lies in S or a neighbor of v lies in S.

(a) Prove or disprove that the greedy algorithm taught in Lecture 3 for computing 2-approximate vertex cover can also compute a 2-approximation of optimal dominating set.

(b) Design a linear time algorithm to compute a dominating set of minimum possible size for trees, and justify the correctness of your algorithm.

Question 5 There are n workshops to be held in technical fest at IIT Delhi. Each workshop is scheduled to operate for a pre-decided single contiguous interval of time. Your task is to allot lecture rooms to different workshops such that no two overlapping workshops get the same lecture room.

(a) Design a greedy algorithm for this problem that uses as few lecture rooms as possible.

(b) How will your solution change if the following additional constraint is imposed - The gap between two consecutive workshops in each lecture room must be at least 30 minutes.