COL 351 : Analysis and Design of Algorithms

Tutorial Sheet - 1

Question 1 Consider the following modified implementation of the Merge Sort algorithm, wherein, 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.

Set $n = \text{length}(A)$ and $K = \lfloor \sqrt{n} \rfloor$;
if $n \leq 5$ **then** Sort the array in Brute-force manner and return;
Store in B_1 the sub-array $A[\,0,\ K]$;
Store in B_2 the sub-array $A[\ K+1, n-1\ ]$;
MergeSort(B_1);
MergeSort(B_2);
Set $x, y, pos = 0$;
while $(x < length(B_1)$ or $y < length(B_2))$ **do**
    if $(B_1[x] \leqslant B_2[y]$ and $x < length(B_1))$ **then**
        set $A[pos] = B_1[x]$, and increment pos and x by 1;
    else set $A[pos] = B_2[y]$, and increment pos and y by 1;
end

Algorithm 1: MergeSort(A)

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

Problem 2 You are given a collection of n jobs $\big(J_1 = (s_1, t_1),\ \ldots,\ J_n = (s_n, t_n)\big)$ and two servers. Provide an algorithm to find a maximum subset of jobs $(J_1, \ldots, J_n)$ that can be scheduled on the two input servers. Also prove the correctness and the time-complexity of your algorithm.

Remark: If job J_i is scheduled on p^{th} server $(p = 1, 2)$, then it occupies that server for the entire time-interval (s_i, t_i) (we do not include in the end-points).

Problem 3 [Kleinberg and Tardos, Chapter 4]
Let's consider a long, quiet country road with houses scattered very sparsely along it. (We can picture the road as a long line segment.) Further, let's suppose that the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four kilometres of one of the base stations.

Give an efficient algorithm that achieves this goal, using as few base stations as possible.

Problem 4 Prove that in an undirected graph, there are even number of vertices of odd-degree.

Problem 5

(a) Let T be a spanning tree of $G = (V, E)$ and $e_0 = (x, y)$ be any edge not lying in T. For any $a, b \in V$, let $\textsc{path}(a, b, T)$ denote the unique path from a to b in T. Then prove that on replacing any edge lying on $\textsc{path}(x, y, T)$ with e_0 we get another spanning tree of G.

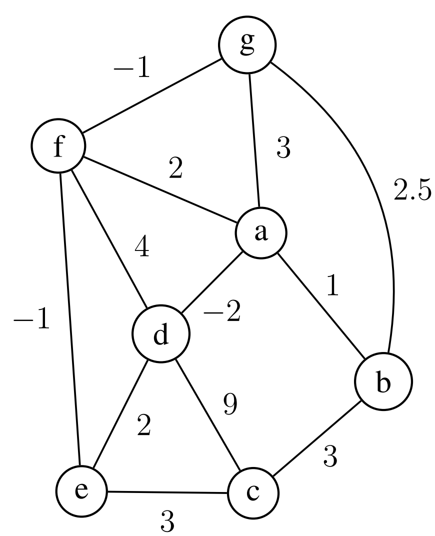
(b) Prove the correctness of the following algorithm to compute MST of a weighted graph $G = (V, E, wt)$.

```
Initialize T to be any arbitrary spanning tree of G.
while there exits e_0 = (x, y) ∉ T and e ∈ PATH(x, y, T) satisfying wt(e_0) ⪇ wt(e) do
    Replace e with e_0 in T;
end
Output T.
```

Algorithm 2: MST($G = (V, E, wt)$)

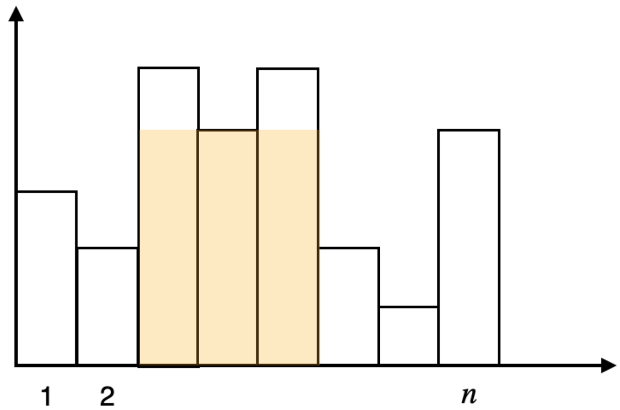
Problem 6

(a) Prove or disprove: There exists an MST of the following graph containing the edge (d, f).



(b) Consider a weighted graph $G = (V, E, wt)$. Prove that an edge $e_0 = (x, y) \in E$ doesn't lie in any MST of G if and only if there exists a path connecting x and y consisting only of edges whose weight is strictly less than $wt(e_0)$.

Challenge Problem 1 Suppose you are given a histogram consisting of n bars of unit length.



Derive an $O(n)$ time algorithm to find the axis-parallel rectangle of maximum area which is covered by the histogram.