COM S 311 Exam 2: Solution Sketch

Not complete and rigorous solutions. Only High level ideas/sketches. In an exam, more rigorous and complete solutions are expected. Complete Solutions were covered in interactive sessions and recitations.

1. **Solve the recurrences (assume $T(2)$ or $T(3)$ is constant). Do not use Master Theorem.**

   (a) $T(n) = 3T(n/2) + n$

   (b) $T(n) = T(n/3) + T(2n/3) + c$

   1. Look at Recurrence tree. Each node has 3 children. Level k (from top) has 3^k nodes each of size $n/2^k$. Wor f levels is $\log_2 n$. Time for last level is dy, this the total time.

   2. Gives an imbalanced tree. Max depth is $log_{3/2}n$. Level k has at most 2^k nodes and time at this level is $2^k c$. Thus total time is $2^{\log_{3/2} nc}$ which is $O(n)$.

2. **Given a directed graph $G = (V, E)$, we say that a vertex $v \in V$ is a bottom vertex if there exists a path from all vertices to v. Write an algorithm to find all bottom vertices in a directed graph. Justify the correctness of your algorithm. Derive t algorithm.**

   DO a DFS. Look at the node with least end time x. T ertex. Now, reverse the graph and and check if every node can be reached from x in the reverse graph. Now do SCC x. Total time is

   Justification: If th here is a bottom vertex x, and say y is not a bottom vertex. Now there is a path from y to x and no path from x to y. Thus by DFS property, end t e smaller then the end time of y. Thus y can not have least end time.

3. **Let $G = (V, E)$ be a weighted undirected graph and let . et $d[u]$ denote the length of the shortest path from s to u. We say that an edge $\langle x, y\rangle$ is *purple edge* if the following condition holds:**

   $$d[y] = d[x] + c(\langle x, y\rangle),$$

   where $c(\langle x, y\rangle)$ is the cost of the edge $\langle x, y\rangle$.

   Give an algorithm that gets G and s as input and outputs all *purple edges*. State the run-time of your algorithm.

   Run Dijkstra and compute the distance array d. Now for every $x \in V$ Do the following: For every (x, u) if $d[u] = d[x] + c(xu)$ output (xu). Time $O(m \log n)$.

4. **Given a sorted array A of integers and an integer k, write an algorithm that outputs the number of times k appears in A. Justify the correctness of your algorithm. Derive the runtime of your algorithm.**

   Use modified Binary search to to find an index i such that $A[i] = k$ and $A[i-1] \neq k$. Use Binary search to find j such that $A[j] = k$ and $A[j+1] \neq k$. Return $(j - i + 1)$.

5. **You own a car repair business, and each morning you have a set of customers whose cars need to be fixed. You can fix only one car at a time. Customer i's car needs t_i time to be fixed. Given a schedule (i.e., an ordering of the jobs), let C_i denote the finishing time of fixing customer i's car. For example, if job j is the first to be done, we would have**

$C_j = t_j$; and if job i is done right after job j, we would have $C_i = C_j + t_i$. Each customer i also has a given weight w_i that represents his or her importance to the business. The happiness of customer i is expected to be dependent on the finishing time of i's job. So the company decides that they want to schedule the jobs to minimize the weighted sum of the completion times, $\Sigma_{i=1}^{n} w_i C_i$.

The company is scheduling the jobs in decreasing order of w_i/t_i. Prove that this strategy gives an optimal solution, i.e., minimizes $\Sigma_{i=1}^{n} w_i C_i$. Use an exchange argument to prove correctness.

Inversion: $w_i/t_i > w_j/t_j$ but j is scheduled before j.

If Optimal schedule does not have inversion, then it is same as greedy.

Suppose Optimal sched ersion:
$w_i/t_i > w_j/t_j$ and j is s
Consider O' where i

Contribution of all cars other than i and j to the sum $w_k t_k$ is the same in both O and O'.

In O, suppose that j starts at time L. In O' i starts at time L.

Contribution of i and j in O:

$$(L + t_j)w_j + (L + t_j$$

Contribution of i and j in O':

Now contributio ntradiction.

6. ***Extra Credit.*** There are n towns in a mountainous county. They are connected by m roadways. The objective of the local government officials is t
of roadways are kept clear through the winter and to mainta

You can view the network of roadways as an undirected graph (V, E), $|V| = n$ and $|E| = m$. Each edge e in this graph is annotated with a number a_e, that gives the altitude of the highest point on the road. We'll assume that no two edges have exactly the same altitude value a_e. The height of a path P in the graph is then the maximum of a_e over all edges e on P. Finally, a path between towns i and j is declared to be winter-optimal if it achieves the *minimum possible height* over all paths from i to j. The local government wants to find a subset $E' \subseteq E$ of roadways such that the subgraph (V, E') is connected and for every pair of towns i and j, the height of the winter-optimal path in (V, E') should be no greater than it is in the full graph $G = (V, E)$. We'll say that (V, E') is a *minimum-altitude connected subgraph* if it has this property.

Prove or disprove that a MST is a minimum-altitude connected subgraph.

Let T be MST that is not minimum-altitide subgraph. There exist two vertices u and v such that P_1 is a path from u to v in T, and there is path P_2 from u to v in G and whose altitude is smaller than altitude of P_2. Now max hight edge e in P_1 has higher height than max height edge e' in P_2. Now the union of P_1 and P_2 will contain a cycle with edges e and e'. By cycle property $e \not\in T$. A contradiction.