CS60007 Algorithm Design and Analysis 2019
Assignment 1

Answer all questions.

1. There are n jobs, with processing times $t_1, \ldots, t_n$ and deadlines $d_1, \ldots, d_n$ respectively. The i-th job needs a contiguous interval of length t_i. In some schedule, let the start and end times of the i-th job be s_i and $f_i := s_i + t_i$ respectively. If $f_i > d_i$, the i-th job is said to incur a *lateness* penalty $L_i := f_i - d_i$. If $f_i \leqslant d_i$, define $L_i := 0$. Our objective is to output a schedule that minimizes the total lateness $\sum_{i=1}^{n} L_i$. Show that the following greedy rule is **not** optimal:

   Schedule the jobs in non-decreasing order of deadlines.

   [10 marks]

2. A vertex cover of an undirected graph $G = (V, E)$ is a set $S \subseteq V$ of vertices that 'covers' every edge, i.e., for every edge $(u, v) \in E$, either $u \in S$ or $v \in S$ (or both). Consider the following greedy algorithm to find a vertex cover of minimum size of an input graph:

   (a) $S \leftarrow \emptyset$;
   (b) If there is no edge that is uncovered, return S;
   (c) Let $(u, v) \in E$ be uncovered. $S \leftarrow S \cup \{u, v\}$;
   (d) Go to step (b).

   Show that there exist graphs with arbitrarily large number of vertices, for which the above algorithm returns a vertex cover of size twice that of an optimal vertex cover. This shows that not only the above greedy rule is sub-optimal, but it can potentially produce solutions that are worse than the optimal solution by a factor of 2.

   [10 marks]

3. Let $G = (V, E)$ be an undirected weighted graph, where the edge weights are distinct and non-negative. Let T be a minimum spanning tree of G. Let p be a shortest path (i.e. a path of minimum weight) from a vertex s to another vertex t in G. Now let us

replace the weight $w(e)$ of each edge e by $w(e)^2$. Is T still a minimum spanning tree of G with respect to the new edge weights? Is p still a shortest path from s to t with respect to the new edge weights? Explain your answers.

[10 marks]

4. An undirected graph on n vertices is said to be *treelike* if it has at most $n + 13$ edges. Design a $O(n)$ time algorithm that, given the adjacency list of a weighted treelike graph G, outputs an MST of G. You may assume that the all edge-weights of G are distinct.

[10 marks]

5. You are given a set D containing at most 10 English words. Think of D as a tiny dictionary. Design an efficient dynamic programming algorithm that, given a string of English letters, finds out if the string can be broken up into a space-separated sequence of dictionary words, and prints "yes" or "no" accordingly. Furthermore, if the answer is "yes", the algorithm should also output one way of breaking up the string into a space-separated sequence of dictionary words.

   Example: $D =${"now", "where", "here"}

   Input: "nowhere"
   output: yes. "now here".

   Input: "nohre"
   output: no.

[10 marks]

6. Suppose you have one machine and a set of n jobs $a_1, a_2, \dots, a_n$ to process on that machine. Each job a_j has a processing time t_j, a profit $p_j \geqslant 0$, and a deadline d_j. The machine can process only one job at a time, and job a_j must run uninterruptedly for t_j consecutive time units. If job a_j is completed by its deadline d_j, you receive a profit p_j, but if it is completed after its deadline, you receive a profit of 0. Give a dynamic programming algorithm to find the schedule that obtains the maximum amount of profit, assuming that all processing times and deadlines are integers between 1 and n. What is the running time of your algorithm?

[10 marks]