Mid-term Exam

Name: _____

Fall 2019, CSCE629 Analysis of Algorithms — Oct. 11, 2019

Texas A&M U — Prof. Fang Song

Instructions (please read carefully before start!)

- This take-home exam contains 8 pages (including this cover page) and 4 questions. Total of points is 90.
- You will have till **10am, Oct. 14, 2019** to finish the exam. You must work on your own, and no collaboration or help from any resources other than those made available in class (lecture notes, texts, homework problems, etc.) is permitted.
- Submit your solutions in PDF on Gradescope before the deadline, either scanned or typeset in LaTeX. If you choose to hand-write and scan, *print out this exam sheet and write your solutions on it*. Do your best to fit your answers into the space provided, and attach extra papers only if necessary. If you typeset in LaTeX, *use the provided TeX file*. No other formats are accepted.
- Your work will be graded on correctness and clarity. Make sure your hand writing is legible. You may opt for the "I'll take 15%" option.
- Don't forget to write your name on top (or update the "\studentname" command in the TeX file)!

Grade Table (for instructor use only)

Question	Points	Score
1	20	
2	25	
3	25	
4	20	
Total:	90	

1. *Short answers*. Answer the following, and briefly justify your answer.

   (a) (5 points) You are working on a divide-and-conquer algorithm that given an input of size n, calls itself recursively on 8 subproblems of size $\lceil n/5 \rceil$. Dividing and combining take time $f(n)$. Can the total running time of the algorithm be $O(n)$?

   (b) (5 points) Rank the following functions by *ascending* order of growth. That is, find an arrangement of $g_1, g_2, \ldots$ such that $g_1(n) = O(g_2(n)), g_2(n) = O(g_3(n)), \ldots$. If two function satisfy $g_i(n) = \Theta(g_j(n))$, they can be in either order. Recall that $\log(\cdot)$ is base 2 logarithm and $\log_b(\cdot)$ is base b logarithm.

$$5^{\log n}, \log^{91} n, \sqrt[5]{n}, 2^{\sqrt[3]{n}}, \log(\sqrt{n!}), 3^{log_3 n}, n^{2\log 7}.$$

   (c) (5 points) Your friend Hulk proposes adapting Dijkstra's algorithm in the following way to deal with negative-length edges. Pick the edge with the smallest length $\ell < 0$ (e.g., -10), and then increase the length of each edge by $|\ell|$ so all edges will become non-negative. Then run Dijkstra's on G with this updated lengths to find a shortest path from s to t. Does this also give you a shortest $s - t$ path in the original graph? Justify your answer.

   (d) (5 points) We have a connected graph $G = (V, E)$. Suppose that we run both BFS and DFS on a vertex $u \in V$, and obtain the same BFS search tree and DFS search tree T, which contains all vertices of G. Prove or disapprove: $G = T$.

2. Consider a function f that takes two integer inputs i, j in $\{1, \ldots, n\}$ and returns a real number. A *local minimum* of f is a point (i, j) such that $f(i, j) \leq f(a, b)$ for all pairs $(a, b) \in \{1, \ldots, n\}^2$ where $|a - i| \leq 1$ and $|b - j| \leq 1$.

   The goal of this problem is to find an efficient algorithm that finds a local minimum of f, assuming nothing about the structure of f except that for any input (i, j), evaluating $f(i, j)$ takes *unit* time.

   We can represent this problem via a grid-like graph G_n, where the vertices are pairs of integers and two pairs are connected if each of their components differs by at most 1, that is $((i, j), (a, b)) \in E$ iff. $|a - i| \leq 1$ and $|b - j| \leq 1$. Note that G_n has n^2 vertices and degree at most 8. We can think of f as a function that assigns a real number to each vertex. A local minimum is then a vertex v such that $f(v) \leq f(v')$ for all adjacent vertices v'.

   (a) (10 points) Give a recursive algorithm for this problem that cuts the graph G_n into four sub-graphs of the form $G_{n/2}$, and gets called recursively on at most one of the subgraphs.

(b) (5 points) Prove correctness of your algorithm.

(c) (10 points) State a recurrence that describes the worst-case running time of your algorithm. Solve it using any method of your choice.

3. You have been hired to plan the locations of rest stops along a stretch of highway. There are n potential locations for the rest stops. Location i is d_i miles from the start point of the highway. Building a rest stop at location i costs c_i dollars. The cost of building at a set of locations is the sum of the individual costs. You need to design an algorithm that takes the numbers (d_i, c_i) and returns the minimum-cost set of locations such that no two consecutive locations are more than 30miles apart, and there is at least one rest stop within 30miles of both the start and end of the highway. Assuming that the input comes sorted in *ascending* order according to d_i.

   (a) (10 points) You realize that dynamic programming is helpful here, assuming that all potential locations are at least 5miles apart. Describe the subproblems for which your algorithm compute solutions.

   (b) (7 points) Give the pseudocode of your algorithm.

(c) (8 points) State your algorithm's (worst-case) running time and justify your answer.

4. (20 points) Suppose you are given a set L of n line segments in the plane, where each segment has one endpoint on the vertical line $x = 0$ and one endpoint on the vertical line $x = 1$, and all $2n$ endpoints are distinct. Describe and analyze an algorithm to compute the largest subset of L in which no pair of segments intersects.

Scrap paper – no exam questions here.