NAME: ______________________________

CSE 421
Introduction to Algorithms
Final Exam Autumn 2021

Anup Rao

December 9

DIRECTIONS:

- Answer the problems on the exam paper.
- You are allowed a single cheat sheet.
- Justify all answers with proofs, unless the facts you need have been proved in class or in the book.
- If you need extra space use the back of a page
- You have 1 hour and 50 minutes to complete the exam.
- Please do not turn the exam over until you are instructed to do so.
- Good Luck!

1	/60
2	/20
3	/20
4	/20
5	/25
Total	/130
Extra	/ 10

1. (60 points, 5 each) For each of the following problems answer **True** or **False** and BRIEFLY JUSTIFY you answer.

   (a) Let (S, T) be a minimum st-cut in a flow graph (G, s, t). If the capacity of every edge in G is increased by 2 then the value of the maximum flow in G is increased by 2 times the number of edges from S to T in G.

   (b) There is an exponential time algorithm for $3SAT$.

   (c) There is a linear time algorithm for finding the $n/10$'th largest number in a list of n numbers.

   (d) If $T(n) = 3T(n/2) + n$ for $n \geq 2$ then $T(n)$ is $\Theta(n^{\log_3 2})$.

   (e) If $T(n) = 12T(n/4) + n^2$ for $n \geq 4$ then $T(n)$ is $\Theta(n^2)$.

(f) If $3SAT$ has a linear time algorithm, then so does every problem in NP.

(g) Every decision problem has an exponential time algorithm.

(h) The average degree of a vertex in a tree with n vertices is exactly 2.

(i) If e is an edge whose weight is lower than the weight of all other edges in an undirected graph, then every minimum spanning tree must contain e.

(j) In a flow network, the value of any valid flow is at most the capacity of any $s - t$ cut.

(k) There is a polynomial time algorithm for Vertex Cover that finds the optimal vertex cover up to a factor of 2.

(l) If the optimal solution to a linear program is 10, then the corresponding dual linear program must have a solution of finite value.

2. (20 points) **Assigning teachers to courses**

The *Teacher Assignment* problem is: given a set of teachers $T = \{t_1, \ldots, t_n\}$ and a set of courses $C = \{c_1, \ldots, c_m\}$ determine an assignment of teachers to courses. The input to the problem has a bipartite graph $G = (T, C, E)$ where an edge (t, c) indicates that teacher t can teach class c. For each teacher t_i there is an integer u_i giving the number of courses that t_i must teach, and for each course c_j there is an integer d_j indicating how many teachers must be assigned to the course. A teacher t can be assigned at most once to course c (in other words, if multiple teachers are required for a course, they must be distinct).

Describe how network flow can be used to find an assignment of teachers to courses. If no assignment is possible that meets these constraints, the algorithm should report failure. Be sure to argue the correctness of your solution.

(You can also use space on the next page for your answer)

3. (20 points) Let $G = (V, E)$ be an undirected graph. Suppose that each edge e has a cost $c(e)$, with $c(e) \in \{1, 2, 3\}$. Describe an $O(n + m)$ time algorithm to compute a minimum spanning tree for G. HINT: Observe that if all the edge costs were exactly the same, then *any* spanning tree would be a minimum spanning tree of the graph.

4. (20 points) Consider the linear program:

$$\begin{aligned} &\text{maximize } x_1 - 2x_3 \\ &\text{subject to} \\ &x_1 - x_2 \leq 1 \\ &2x_2 - x_3 \leq 1 \\ &x_1, x_2, x_3 \geq 0 \end{aligned}$$

Prove that $(x_1, x_2, x_3) = (3/2, 1/2, 0)$ is an optimal solution.

5. (25 points) The following problem can be useful for data compression. The input is a string y of length n and a list of k strings $x_1, \ldots, x_k$ of lengths $m_1, \ldots, m_k$ respectively, where each x_i is represented as an array $x_i[1] \cdots x_i[m_i]$ and y is $y[1] \cdots y[n]$. The problem is to determine the smallest number of copies of strings from $x_1, \ldots, x_k$ that can be concatenated together to produce y. For example, if $x_1 = a$, $x_2 = ba$, $x_3 = abab$, and $x_4 = b$ and $y = bababbaababa$ then we can write $y = x_4 x_3 x_2 x_3 x_1$ which is optimal, so the answer is 5.

   (a) (10 points) For $i \geq 0$, let $Opt(i)$ be the optimal number of strings required to produce the string $y[1] \cdots y[i]$. Describe a recursive algorithm for computing $Opt(n)$. Don't forget the base case.

   (b) (10 points) Describe how you can compute $Opt(n)$ efficiently using an iterative dynamic programming algorithm.

   (c) (5 points) What is the running time of your algorithm in terms of n, k and $m = \max_i m_i$?

(You can also use space on the next page for your answer)

6. (10 points Extra Credit) Give an $O(n^3)$ time algorithm for finding a 4-cycle in an undirected graph, if one exists. HINT: If a graph has a 4-cycle $abcd$, then there must be two distinct paths of length two between a and c. For every vertex b, keep track of the paths of length 2 that are generated with b in the middle.