CSC 425/520 A01 FALL 2020 (CRN: 10828 and 10839)
ANALYSIS OF ALGORITHMS
FINAL EXAMINATION
UNIVERSITY OF VICTORIA

1. Student ID: ________________________

2. Name: ____________________________

3. DATE: 16 DECEMBER 2020
   DURATION: 3 HOURS
   INSTRUCTOR: V. SRINIVASAN

4. THIS QUESTION PAPER HAS TEN PAGES INCLUDING THE COVER PAGE.

5. THIS QUESTION PAPER HAS NINE QUESTIONS.

6. READ THROUGH ALL THE QUESTIONS AND ANSWER THE EASY QUESTIONS FIRST. KEEP YOUR ANSWERS SHORT AND PRECISE.

Q1 (4)	
Q2 (6)	
Q3 (6)	
Q4 (4)	
Q5 (6)	
Q6 (6)	
Q7 (6)	
Q8 (6)	
Q9 (6)	
TOTAL (50) =	

1. **Stable Matching**

   Consider a town with n men and n women seeking to get married to one another. Each man has a preference list that ranks all the women, and each woman has a preference list that ranks all the men. Men propose to women and women decide whether to accept the proposal or not.

   The set of all $2n$ people is divided into two categories: good people and bad people. Suppose that for some number k, $1 \leq k \leq n-1$, there are k good men and k good women; thus there are $n-k$ bad men and $n-k$ bad women. Everyone would rather marry any good person than any bad person. Formally, each preference list has the property that it ranks each good person of the opposite gender higher than each bad person of the opposite gender: its first k entries are the good people (of the opposite gender) in some order, and its next $n-k$ are the bad people (of the opposite gender) in some order.

   Prove that in every stable matching, every good man is married to a good woman. [4 Marks]

2. **Greedy Algorithms**.

   (a) Consider the *interval scheduling problem* in which we have set of jobs $\{1, 2, \ldots, n\}$ and the start and finish times of job i are denoted by s_i and f_i. A subset of jobs are said to be *compatible* if they do not overlap. The goal in this problem is to find the maximum size subset of compatible jobs. Give a counter-example to show that a greedy algorithm that schedules jobs based on *earliest start time* cannot find the maximum size subset of compatible jobs. [3 Marks]

   (b) Let S be any finite set and for an integer k, $I = \{Y \subseteq S \mid |Y| \leq k\}$. Then (S, I) is.a matroid. Is this true or false? Explain why. [1.5 Marks]

   (c) Let $G = (V, E)$ be an undirected graph. Recall that subset $Y \subseteq E$ is a matching if each vertex of V is contained in at most one of the edges of Y. Let $S = E$ and Let $I = \{Y \subseteq S \mid Y \text{ is a matching}\}$. Then (S, I) is a matroid. Is this true or false? Explain why. [1.5 Marks]

3. **Divide and Conquer.**

Suppose that we are given a coefficient representation of a univariate polynomial $A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{n-1}x^{n-1}$ and we would like to compute its point-value representation $A(x) : (x_0, y_0), \ldots, (x_{n-1}, y_{n-1}$.

(a) What would be the running time of the brute force method? Explain why. [2 Marks]

(b) What is the running time of this procedure when FFT is used? Explain the main idea behind this method by showing how to evaluate a polynomial of degree $n-1$ at 4 points by evaluating two polynomials of degree $\frac{n}{2} - 1$ at only two points. [4 Marks]

4. **Dynamic Programming Algorithms.**

   Recall the dynamic programming algorithm for sequence alignment we saw in the class. Using that algorithm, compute the edit distance between STALL and TABLE. Assume a gap penalty = 1 and mismatch penalty =3. Show your computation by building the table M as in slide 10 of the lecture slides on DP. [4 Marks]

5. **NP-Hardness**

   Given an unweighted, directed graph $G = (V, E)$, a path $\langle v_1, v_2, ..., v_n \rangle$ is a set of vertices such that for all $0 < i < n$, there is an edge from v_i to v_{i+1}. A cycle is a path such that there is also an edge from v_n to v_1. A simple path is a path with no repeated vertices and, similarly, a simple cycle is a cycle with no repeated vertices. In this question we consider two problems:

   - LONGEST-SIMPLE-PATH: Given a graph $G = (V, E)$ and two vertices $u, v \in V$, find a simple path of maximum length from u to v or output NONE if no path exists.
   - LONGEST-SIMPLE-CYCLE: Given a graph $G = (V, E)$, find a simple cycle of maximum length in G.

   Assuming that the problem of finding the longest simple path is NP-hard, show that the problem of finding the longest simple cycle is NP-hard. To do so, give a polynomial time reduction from the problem of finding the longest simple path to the problem of finding the longest simple cycle. Prove the correctness of your reduction. [6 Marks]

6. **Tractability of NP-hard Problems for Special Cases.**

   In the class, we saw a greedy algorithm for computing independent sets in trees. The key observation was that if v is a leaf, then there is a maximum size independent set containing v. Modify this algorithm to design a greedy algorithm for computing minimum size vertex covers in trees. Give a proof of correctness. (**Hint: Prove that for any leaf node v, there is a minimum size vertex cover not containing v.**) [6 Marks]

7. **Approximation Algorithms I.**

   Consider the Max-2-SAT problem defined as follows: Given a CNF formula ϕ in which each clause contains exactly two literals, find a truth assignment for the variables that satisfies as many clauses as possible. Formulate the Max-2-SAT problem as an ILP. (**Hint: Assign an indicator for each variable and each clause.**) [6 Marks]

8. **Approximation Algorithms II.**

   Consider the following problem of computing maximum size matching in bipartite graphs. Given disjoint sets of vertices X and Y, and given a set $T \subseteq X \times Y$ of edges, a subset $M \subseteq T$ is a matching if each vertex of $X \cup Y$ is contained in at most one of these edges. The Maximum Size Matching Problem is to find a matching M of maximum size. (The size of the matching is the number of edges it contains. You may assume $|X| = |Y|$ if you want.) Give a polynomial-time algorithm that finds a matching of size at least $1/2$ times the maximum possible size. Give a proof of correctness. [6 Marks]

9. **Randomized Algorithms.**

   Suppose that we are given a set of n variables $x_1, x_2, \ldots x_n$ each of which can take values 0 or 1. We are also given a set of k equations where the rth equation has the form

$$(x_i + x_j + x_k) \bmod 2 = b_r$$

   for some choice of three distinct variables x_i, x_j, x_k and some value b_r is 0 or 1. Thus each equation specifies if the sum of three variables is odd or even. Consider the problem of finding an assignment to the variables that maximizes the number of equations satisfied. Design a polynomial time randomized algorithm that outputs an assignment that satisfies 1/2 fraction of the equations, Give a proof of correctness. (**Hint: Think about how the randomized algorithm for Max 3-SAT was designed in the class)**. [6 Marks]

END OF EXAM