CSE 551 - Foundations of Algorithms
Mid Term, Fall 2022
Closed Books, Closed Notes
Time: 1 hour
Answer any three questions
Each question carries 25 pts.

Important Note: *For the purpose of grading your answers, significant emphasis will be given to the process through which you arrive at the answer. In other words, points will be deducted if you do not provide proper justification for your answer, even when you provide the correct answer. If the question asks for finding the answer following a specific technique, points will be deducted if you don't follow that technique but still provide the correct answer.*

Problem 1: Suppose there is a set A of men and a set B of women. Each set contain n elements. There exist two $n \times n$ arrays P and Q such that $P(i, j)$ is the preference of man i for woman j and Q is the preference of woman i for man j . Give an algorithm which finds a pairing of men and women such that the following condition is **not satisfied**. *There is an element $a_i \in A$ that has a higher preference for an element $b_k \in B$ over the element $b_j \in B$ with which a_i is paired, and $b_k \in B$ has a higher preference for $a_i \in A$ over the element $a_l \in A$ with which b_k is paired.* Prove the correctness of your algorithm (i.e., it ensures that the given condition isn't satisfied).

Problem 2: Compute the best case and worst case complexity of the following algorithm. Show all your work.

Algorithm XYZ(S)

if $|S| = 2$ then compare the two numbers and return (min,max)

else

begin

1. Pick an arbitrary element s_k of the sequence S.
2. Divide S into parts S_1, S_2,S_3 such that the elements of S_1, S_2, and S_3 are less than, equal to and greater than s_k respectively.
3. return (XYZ(S_1),S_2,XYZ(S_3))

end

Problem 3: Using Dynamic Programming technique, find the optimal solution to the Traveling Salesman Problem for the folowing data set (Distance Matrix).

$$M = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 20 \end{bmatrix}$$

Show all your work. Also show the node ordering in the optimal tour.

Problem 4: Consider the following job scheduling problem. Given a set of jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$. Each job has a specific *start time,* s_i and a specific *finisht time,* f_i. Also given a set of processors $\mathcal{P} = \{P_1, \ldots, P_n\}$. Processing cost of one job in processor P_i is C_i, $1 \leq i \leq n$. Processing costs are in *increasing order*, i.e., $C_1 \leq C_2 \leq \ldots \leq C_n$. If two jobs have overlapping time requirement, they cannot be scheduled on the same processor. If $n = 10$ and the jobs are assigned to the processors in a way that the P_1 processes 5 jobs, P_2 processes 3 jobs and P_3 processes 2 jobs, then the *total processing cost* is $5C_1 + 3C_2 + 2C_3$. The goal of this job scheduling problem is to assign jobs to the processors in a way that *minimizes* the total processing cost.

Suppose that the following strategy is used to schedule the jobs on the processors. A set of jobs are said to be *mutually compatible*, if they do not have any overllaping time *conflict*. The strategy first computes the *largest set of mutually compatible* jobs and assigns them to the *cheapest* processor. It then removes this set of jobs from consideration and among the remaining set of jobs finds the largest set of mutually compatible jobs and assigns them to the second *cheapest* processor. This process continues till all the jobs are assigned to a processor.

Is this strategy always going to minimize the total processing cost? If your answer is "yes" then provide arguments to substantiate your claim. If your answer is "no" then provide an example where this strategy fails to minimize the total processing cost.