ECS 122A, Final exam practice

First name : ________________	Student ID : ________________
Last name : ________________	Additional papers : ____________

Question:	1	2	3	4	5	Total
Points:	6	9	6	6	8	35
Score:						

Answer the questions in the spaces provided. If you run out of room for an answer, continue on the back of the page. If you still run out of room, ask for additional papers.

Instructions:

- The final exam lasts **110 minutes**.
- It is **suggested**, although not mandatory, not to use a pencil or an erasable pen.
- Sign each additional page you hand in.
- You can use any algorithm we saw in the class as a black-box. However, if you use it, you are expected to state its properties and running time where needed.

Good luck!

1. Please answer the following questions.
   (a) (2 points) State whether the following statement is **True** or **False** (no justification needed): For the Interval scheduling problem you saw in class, it is always the optimal strategy to include the shortest interval in the solution.

   ☐ True ☐ False

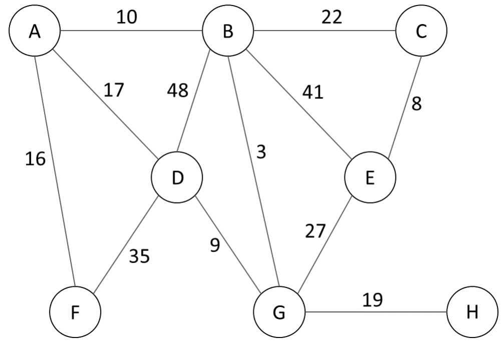
   (b) (2 points) State whether the following statement is **True** or **False** (no justification needed): On two input strings of lengths n and m, the sequence alignment dynamic programming solution you saw in class runs in time $O(n + m)$.

   ☐ True ☐ False

   (c) (2 points) State whether the following statement is **True** or **False** (no justification needed): Given an unweighted connected graph G, Prim's and Kruskal's algorithms executed on G must output the same spanning tree.

   ☐ True ☐ False

2. (a) (3 points) For this graph



simulate Dijsktra's algorithm from the vertex A. Write **only** shortest distances from A to every node computed by this algorithm. Fill in the following table.

Node	Distance from A
B	
C	
D	
E	
F	
G	
H	

(b) (3 points) What is the output of Cashier's algorithm on the following denominations and target value? (As a reminder, the Cashier's algorithm outputs the multiset of denominations used.) Fill in the following table.

- Denominations $= \{19, 13, 4, 1\}$.
- Target value $= 37$.

Denomination	19	13	4	1
# of coins				

(c) (3 points) Consider two strings X = "UCD" and Y = "DUCK". If X and Y are given as the input, the optimal sequence alignment algorithm you saw in the lecture will compute a function OPT, where $OPT(i, j)$ is defined as the minimum cost of aligning the prefix strings $X[1 \ldots i]$ and $Y[1 \ldots j]$ for $0 \leq i \leq |X|$ and $0 \leq j \leq |Y|$. (When $i = 0$, the prefix strings $X[1 \ldots i]$ and $Y[1 \ldots i]$ are defined as the empty string.) Please simulate the algorithm on X and Y by completing the following table.

i \ j	0	1 (D)	2 (U)	3 (C)	4 (K)
0					
1 (U)					
2 (C)					
3 (D)					

Detailed instructions:

- Please write down the value of $OPT(i, j)$ in the cell at the i^{th} row and j^{th} column. To help you compute the table, we mark each row i with $X[i]$ and each column j with $Y[j]$ (for $i, j > 0$). This should help you determine the cost of matching $X[i]$ and $Y[j]$.
- Please use the following parameters:

$$\delta = 2 \text{ and } \alpha_{X_i Y_j} = \begin{cases} 3, & \text{if } X_i \neq Y_j \\ 0, & \text{if } X_i = Y_j \end{cases}$$

  Recall that δ is the cost of matching a character with a gap, and $\alpha_{X_i Y_j}$ is the cost of matching X_i and Y_j.

3. (6 points) Let $G = (V, E)$ be a directed graph with nodes $v_1, v_2, \ldots, v_n$. We say that G is an *ordered graph* if it has the following properties.

   1. Each edge goes from a node with a lower index to a node with a higher index. That is, every directed edge has the form (v_i, v_j) with $i < j$.
   2. Each node except v_n has at least one edge leaving it. That is, for every node v_i, $i = 1, 2, \ldots, n-1$, there is at least one edge of the form (v_i, v_j).

   Each edge e in G is associated with a length ℓ_e. The length of a path is the total length of the edges in the path. Note that the edges of G are directed. That is, an edge (v_i, v_j) can only be used to travel from v_i to v_j (and not from v_j to v_i). The goal of this question is to solve the following problem.

   Given an ordered graph G and the length of each of its edge, find the length of the ***shortest*** *path that begins at v_1 and ends at v_n.*

   Your algorithm should run in $O(n+m)$ time. Figure 1 gives an example in which the length of each edge is labeled beside it. The shortest path is (v_1, v_3, v_5, v_6). The length of this path is $5+5+3 = 13$. Note that $(v_1, v_3, v_5, v_4, v_6)$ is not a path, because the edge (v_4, v_5) cannot be traveled in the reverse direction.

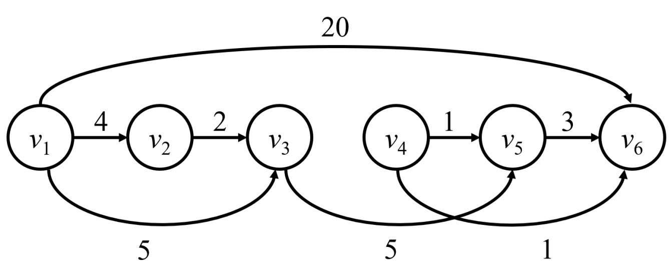


Figure 1: Example of an ordered graph G.

4. (6 points) You are given a graph G on n vertices and n edges, i.e., $m = n$. There are no self-loops nor parallel edges. Also, G is not necessarily connected. Design an algorithm that determines whether it is possible to remove one edge to make the graph acyclic, and if so, outputs one such edge. Your algorithm should run in $O(n)$ time. For the full score, analyze its running time and comment on the correctness. Informal and brief, but **informative**, arguments for correctness will also score full points.

   Remark: If you do not submit a solution to this problem, i.e., if your solution is left blank, you will score **3 points**. Incorrect solutions might score **less** than 3 points.

5. (8 points) Design an algorithm for the *regular expression matching problem*, described as follows. You are given two strings T and P, called the *text string* and *pattern string*, respectively. The text string contains only English letters. The pattern string may contain English letters and two types of special characters: the dot character (“`.`”) and the star character (“`*`”). A star character can match a (**possibly empty**) substring of the text, and a dot character can match **exactly one** English letter of the text. Your task is to determine whether the text matches the pattern. More formally, you have to design an algorithm that determines whether it is possible to transform the pattern string into the text string using the following two operations an arbitrary number of times:

   - Replace a star character with a (possibly empty) string of English letters.
   - Replace a dot character with exactly one English letter.

   To be considered correct, your algorithm should run in $O(|T| \cdot |P|)$ time. Some examples are given below.

   - If $T = P =$ “`DavisTheBest`”, your algorithm should output yes.
   - If $T =$ “`DavisTheBest`” and $P =$ “`Dav*The*`”, your algorithm should output yes, because you can replace the first star by “`is`” and the second star by “`Best`”.
   - If $T =$ “`DavisTheBest`” and $P =$ “`Davi.The*Bes.`”, your algorithm should output yes. Note that the star is replaced by an empty string.
   - If $T =$ “`DavisTheBest`” and $P =$ “`DavisTheBest.`”, your algorithm should output no, because the last dot must match exactly one English letter.

   For the total score, analyze the running time of your solution and provide the intuition behind it.

   Remark: If you do not submit a solution to this problem, i.e., if your solution is left blank, you will score **3 points**. Incorrect solutions might score **less** than 3 points.