Midterm 2

Name:

SID:

GSI and section time:

Write down the names of the students on your left and right as they appear on their SID.

Name of student on your left:

Name of student on your right:

Name of student behind you:

Name of student in front of you:

Instructions:

Write your name and SID on each sheet in the spaces provided.

You may consult two handwritten, double-sided sheets of notes. You may not consult other notes, textbooks, etc. Cell phones and other electronic devices are not permitted.

There are 5 questions. The last page is page 12.

Answer all questions. On questions asking for an algorithm, make sure to respond in the format we request. Write in the space provided. Good luck!

Note: If you finish in the last 15 minutes, please remain seated and do not leave early, to avoid distracting your fellow classmates.

Do not turn this page until an instructor tells you to do so.

1. (12 pts.) Dynamic Warmup

(a) The recurrence of a dynamic programming algorithm is

$$C[i] = \min\{C[i-1] + a_i, C[i-2] + b_i\},$$

initialized to $C[0] = C[1] = 0$. a_i and b_i, for $i \in 1, \ldots, n$, are constants. The desired answer is $C[n]$. Write full *iterative* pseudocode for this algorithm; do not use recursion or memoization. (For convenience, you can assume $C[0]$ and $C[1]$ have been set to 0).

(b) Fill in the edit distance table for the words *precise* and *practice*. We've already started it off for you. Recall that entry (c, r) in an edit distance table contains the edit distance between the length-c prefix of the column word and the length-r prefix of the row word. For example, $E(3,1) = 2$, representing the edit distance between *pre* and *p*.

Then, circle the subproblems that are used to get from the base case $(0,0)$ to the solution.

		P	R	E	C	I	S	E
	0	1	2	3	4	5	6	7
P	1	0	1	2	3	4	5	6
R	2	1	0	1	2	3	4	5
A	3	2	1					
C	4	3	2					
T	5	4	3					
I	6	5	4					
C	7	6	5					
E	8	7	6					

Name: ______________________ SID: __________

2. (18 pts.) Linear Programming Potpourri

(a) Call the linear program below LP_1. What is the dual?

$$\begin{aligned} \max \quad & x+y \\ \text{subject to:} \quad & x+3y \leq 20 \\ & 2x+y \leq 10 \\ & x, y \geq 0 \end{aligned}$$

(b) What is the optimum solution of LP_1? Give multipliers for the two inequalities of LP_1 which show that this is indeed the optimum value.

x: ☐

y: ☐

optimum value:

☐

multiplier for $x+3y \leq 20$:

☐

multiplier for $2x+y \leq 10$:

☐

(c) Start from $(0,0)$ and perform one step of simplex in LP_1. What are the new equations?
Alternatively, choosing to increase y, we move to $(0,20/3)$.
The first constraint is tight so we set $y' = 20-3y-x$ and $x' = x$. Solving for y, we have $y = 20/3 - x'/3 - y'/3$. Substituting into LP_1, we get the new LP.

$$\begin{aligned} \max \;\; & 20/3 + 2x'/3 - y'/3 \\ \text{subject to:} \;\; & 5x' - y' \leq 10 \\ & x' + y' \leq 20 \\ & x', y' \geq 0 \end{aligned}$$

(d) What is the value of this two-player zero-sum game? The values in the matrix indicate outcome given the corresponding actions by Row and Column; Row is the maximizer and Column is the minimizer. Briefly justify your answer.

		Column: Horse	Dice
Row:	Horse	1	0
	Dice	0	1

value:

(e) You found a way to multiply 4×4 matrices with only M multiplications, and this gives rise to a divide-and-conquer algorithm that is faster than Strassen's algorithm! What is the largest value of M for which this is true? Justify briefly.
Recall that Strassen's algorithm multiplies an $n \times n$ matrix with 7 multiplications of $n/2 \times n/2$ matrices.

value of M:

3. (28 pts.) True/False

Write T or F in the box.
Then, if the statement is true, justify briefly. If the statement is false, give a simple *counterexample.*

(a) ☐ In a Huffman code, if there is a leaf of depth 1 and a leaf of depth 3, then there must also be a leaf of depth 2.

(b) ☐ In a Huffman code where all letter frequencies are distinct, the third least frequent letter has depth either equal to that of the two least frequent letters, or one less.

(c) In an instance of SET COVER, we are given 100 elements to cover by a family of 10,000 sets. It is possible for the greedy solution to be 6 times the optimum. ($\ln 100 \approx 4.6, \ln 10,000 \approx 9.3$.)

For this question, justify briefly whether you write True or False, rather than giving a counterexample.

(d) If a linear program is unbounded, then its dual must be unbounded.

(e) If the capacities of a max flow problem are multiples of $\frac{1}{2}$, the value of the optimum flow will also be a multiple of $\frac{1}{2}$.

(f) $\square$ If we add an integer $k > 0$ to all capacities of a max-flow problem, the max flow is also increased by exactly k.

(g) $\square$ If we multiply all capacities of a max flow problem by an integer $k > 0$, the max flow is also multiplied by exactly k.

4. (12 pts.) Supercomputer Scheduling

You must schedule the operation of a supercomputer for the next n days. For each day $i \in 1, \ldots, n$, you know c_i, the number of heptacycles needed that day. The supercomputer must be rebooted after at most k consecutive days of work. It takes one day to reboot the supercomputer, and the jobs for that day are outsourced with a cost of $1 per heptacycle. Assume that the machine was rebooted at day 0. Find the rebooting schedule that minimizes cost, which is the number of outsourced heptacycles.

For example, if $n = 11$, $k = 3$, and the c_i's are $[7,8,1,8,6,6,3,2,3,6,5]$, the optimum schedule would be to reboot at days 3, 7, and 8, at a total cost of 6.

We'll solve this problem with DP, with the following subproblems: For $i \in 1, \ldots n$, define $Q[i]$ to be the minimum cost possible for scheduling the first i days if the machine is **not** rebooted on day i. Similarly, define $R[i]$ to be the minimum cost if the machine **is** rebooted on day i.

Complete the pseudocode below.

$Q[0] = R[0] =$ ____________________

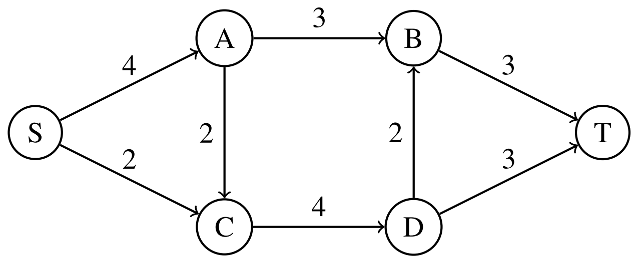
for $i = 1, \ldots, n$:

$Q[i] = \min$ ____

$R[i] = \min$ ____

optimum cost = ____

5. (25 pts.) Fantastic Flows



(a) Answer the following questions about the max flow from S to T through the graph above. No justifications are necessary.

Size of max flow:

Find a minimum cut
(specify the two sets of vertices which define it):

Is the minimum cut unique?

(b) Can the maximum flow in the above network be found with three or fewer iterations of the Ford-Fulkerson algorithm? Explain briefly.

Parts (c) and (d) refer to general max flow networks, not specifically the network on the previous page.

(c) Complete the following sentence (justify briefly, assume all capacities are integers):

Decreasing the capacity of an edge by 1 decreases the flow if and only if the edge belongs to a ...

(d) Complete the following sentence (justify briefly, assume all capacities are integers):

Increasing the capacity of an edge by 1 increases the flow if and only if the edge belongs to ...

(e) Give a polynomial-time algorithm for determining, given a flow network, whether it has a unique minimum cut. You can describe your algorithm without pseudocode if you prefer. Briefly justify correctness and polynomial running time; assume all capacities are integers.

Blank Space

You can use this extra space for scratch work or continuing to answer a question.