Midterm 2

- **The exam will last 110 minutes.**
- ***The answer boxes are for in-person exams. You do not need to print out the exam and answer in "boxes".*** Make sure you clearly mark your answers.
- The exam has 8 questions with a total of 100 points. You may be eligible to receive partial credit for your proof even if your algorithm is only partially correct or suboptimal.
- Answer all questions. Read them carefully first. Not all parts of a problem are weighted equally.
- Be precise and concise.
- The problems may **not** necessarily follow the order of increasing difficulty.
- The points assigned to each problem are by no means an indication of the problem's difficulty.
- Unless the problem states otherwise, you should assume constant time arithmetic on real numbers.
- If you use any algorithm from lecture and textbook as a blackbox, you can rely on the correctness and time/space complexity of the quoted algorithm. If you modify an algorithm from textbook or lecture, you must explain the modifications precisely and clearly, and if asked for a proof of correctness, give one from scratch or give a modified version of the textbook proof of correctness.
- Good luck!

1 Huffman Warmup

(4pts) Given a Huffman encoding of a file F, how would you produce a Huffman encoding of the file $F||F$ (the concatenation of the file and itself)? Justify your answer.

.

2 Flow Computation

(8pts) Find the maximum s-t flow and minimum s-t cut on the graph below. (Fill in the appropriate data in the lines below.)

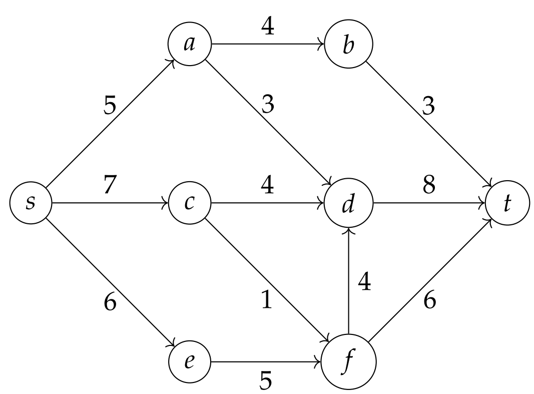

Figure 1: All parts of this problem are with respect to the above graph.

(a) Final max flow and min cut values.

Max flow value: ☐ **Min cut value:** ☐

(b) Final flow across each edge (each blank should be a number):

(s,a)	
(s,c)	
(s,e)	
(a,b)	
(a,d)	
(c,d)	

(c,f)	
(e,f)	
(b,t)	
(d,t)	
(f,d)	
(f,t)	

(c) Edges that cross the min-cut: ☐

3 Flow Proof

Let s, v, t be vertices in G. Prove that if there's a flow of value[1] k from s to v and a flow of value k from v to t in G, then there's a flow of value k from s to t in G.

(8pts) Please complete this elegant five-sentence proof.

Proof. Suppose, for the sake of contradiction, that ____(1)____.
Then, by the ____(2)____ theorem, we know there exist a (s,t)-cut[2] (S,T) of the graph, such that ____(3)____.
Without loss of generality, let $v \in S$.[3]
Since ____(4)____, then by the same theorem, we know that ____(5)____.
Contradiction. □

4 Polytope Park Rangers

(5pts) Consider the following LP.

$$\begin{aligned} &\max 3x + y \\ \text{subject to } &-2x + y \geq -15 \\ &x \geq 2y \\ &y \geq 0 \end{aligned}$$

What is the optimal x, y and objective? Answer in the boxes below.

x^*: [ ], y^*: [ ]

Optimal objective value: [ ]

[1] *i.e.* a size-k flow.
[2] Recall the definition of an (s,t)-cut: a partition of the graph into two sets (S,T) such that $s \in S$, $t \in T$.
[3] In other words, a similar argument holds when $v \in T$.

5 Memory Usage

(5pts) Tianchen is running a multiplicative weight update algorithm on an old computer and he notices his memory usage is at capacity from having too much data while running his MWU program. Which action(s) will definitely decrease memory usage by keeping track of less data when running the MWU program?
A. Decrease number of timesteps.
B. Decrease number of experts.
C. Increase ϵ.
D. Decrease ϵ.
Justify your answer. *i.e.* Explain why the action(s) you chose works and then briefly explain (in one or two sentences) why the other action(s) won't work.

6 Game

(20pts) Consider the following matrix game. A positive payoff goes to the row player. $a, b, c, d > 0$.

$$\begin{bmatrix} a & -b \\ -c & d \end{bmatrix}$$

(a) What is the optimal strategy for the row player? For the column player? Show your work. Remember to justify why your answers are the optimal strategies.

(b) Under what conditions is the game fair; that is, the expected value to the optimal players are 0? Show your work.

7 Nice numbers

(20pts) The problem below is a dynamic programming problem and should be viewed as having 3 parts. You should find a function f which can be computed recursively so that evaluation of f on a certain input (or combining its evaluations on a few inputs) gives the answer to the stated problem.

- Part (1) is to define f *in words* (without mention of how to compute it recursively). You should clearly state how many parameters f has, what those parameters represent, what f evaluated on those parameters represents, and how you should use f to get the answer to the stated problem.
- Part (2) is to give a recurrence relation showing how to compute f recursively, including a description of the base cases.
- In part (3) you should give the running time **and space** for solving the original problem using computation of f via memoization or bottom-up dynamic programming. If you need to use certain data structures, compute in a certain order, preprocess the data in a certain way, *etc.*, to optimize computation of f, you should say so.

Note: if there are multiple solutions to solve the stated dynamic programming problem, you should describe the most time-efficient one you know. If there are multiple solutions with the same asymptotic time complexity, you should describe the implementation that gives the best asymptotic space complexity.

The problem: A number is “nice” if each pair of adjacent digits in its decimal representation differ by at least 2. For example, “13” is nice, but “28549” isn't (the 5 and 4 only differ by one). Given k, calculate the number of nice integers x that are k digits long. That is, calculate the number of nice x such that $10^{k-1} \le x \le 10^k - 1$. You should output this number modulo $P = 1000003$.

8 We're Going To Be Friends

(30pts) Jack lives in Yaweno, where all the expressways between cities are one-way (*i.e.* directed edge). There are n cities and m expressways. Jack has c_i friends living in city i, $i \in \{1, \ldots, n\}$. Jack lives in city s. Jack wants to go to city t. $s, t \in \{1, \ldots, n\}$. Given n, s, t and the m expressways between the n cities, Jack would like to calculate the maximum number of friends he can visit driving from s to t using only expressways.

(a) **For this subpart only**, suppose the expressways of Yaweno **do not form any cycle** (*i.e.* they form a directed acyclic graph). Give a **three-part dynamic programming solution**. The template is copied below for your convenience.

This problem is a dynamic programming problem and should be viewed as having 3 parts. You should find a function f which can be computed recursively so that evaluation of f on a certain input (or combining its evaluations on a few inputs) gives the answer to the stated problem.

- Part (1) is to define f *in words* (without mention of how to compute it recursively). You should clearly state how many parameters f has, what those parameters represent, what f evaluated on those parameters represents, and how you should use f to get the answer to the stated problem.
- Part (2) is to give a recurrence relation showing how to compute f recursively, including a description of the base cases.
- In part (3) you should give the running time **and space** for solving the original problem using computation of f via memoization or bottom-up dynamic programming. If you need to use certain data structures, compute in a certain order, preprocess the data in a certain way, *etc.*, to optimize computation of f, you should say so.

Note: if there are multiple solutions to solve the stated dynamic programming problem, you should describe the most time-efficient one you know. If there are multiple solutions with the same asymptotic time complexity, you should describe the implementation that gives the best asymptotic space complexity.

(b) Suppose you have access to a correct and time-optimal implementation of the special-case version of the problem in part (a) that runs in $O(f(n, m))$. Design an efficient algorithm that solves the general-case version. Describe your algorithm, justify its correctness, and then analyze its runtime in terms of n, m, and $f(n, m)$.