Final

Name:

SID:

Name and SID of student to your left:

Name and SID of student to your right:

Exam Room:

☐ Wheeler 150 ☐ Haas Faculty Wing F295 ☐ Lewis 100 ☐ McCone 141 ☐ Evans 100
☐ Wozniak Lounge ☐ other

Please color the checkbox completely. Do not just tick or cross the box.

Rules and Guidelines

- **The exam is out of 170 points and will last 170 minutes.** Roughly, one should expect to spend a minute for a point.
- Answer all questions. Read them carefully first. Not all parts of a problem are weighted equally.
- Write your student ID number in the indicated area on each page.
- Be precise and concise. **Write in the solution box provided.** You may use the blank page on the back for scratch work, but it will not be graded. Box numerical final answers.
- The problems may **not** necessarily follow the order of increasing difficulty. ***Avoid getting stuck on a problem.***
- Any algorithm covered in lecture can be used as a blackbox. Algorithms from homework need to be accompanied by a proof or justification as specified in the problem.
- Good luck!

Discussion Section

Which of these do you consider to be your primary discussion section(s)? Feel free to choose multiple, or to select the last option if you do not attend a section. **Please color the checkbox completely. Do not just tick or cross the boxes.**

☐ Antares, Tuesday 5 - 6 pm, Mulford 240

☐ Kush, Tuesday 5 - 6 pm, Wheeler 224

☐ Arpita, Wednesday 9 - 10 am, Evans 3

☐ Dee, Wednesday 9 - 10 am, Wheeler 200

☐ Gillian, Wednesday 9 - 10 am, Wheeler 220

☐ Jiazheng, Wednesday 11 - 12 am, Cory 241

☐ Sean, Wednesday 11 - 12 am, Wurster 101

☐ Tarun, Wednesday 12 - 1 pm, Soda 310

☐ Jerry, Wednesday 1 - 2 pm, Wurster 101

☐ Jierui, Wednesday 1 - 2 pm, Etcheverry 3113

☐ Max, Wednesday 1 - 2 pm, Etcheverry 3105

☐ James, Wednesday 2 - 4 pm, Dwinelle 79

☐ David, Wednesday 2 - 3 pm, Barrows 140

☐ Vinay, Wednesday 2 - 3 pm, Wheeler 120

☐ Julia, Wednesday 3 - 4 pm, Wheeler 24

☐ Nate , Wednesday 3 - 4 pm, Evans 9

☐ Vishnu, Wednesday 3 - 4 pm, Moffitt 106

☐ Ajay, Wednesday 4 - 5 pm, Hearst Mining 310

☐ Zheng, Wednesday 5 - 6 pm, Wheeler 200

☐ Neha, Thursday 11 - 12 am, Barrows 140

☐ Fotis, Thursday 12 - 1 pm, Dwinelle 259

☐ Yeshwanth, Thursday 1 - 2 pm, Soda 310

☐ Matthew, Thursday 2 - 3 pm, Dwinelle 283

☐ Don’t attend Section.

1 Zero-Sum Games (5 points)

Consider a zero-sum game given by the following matrix (indicating the payoffs to the row player). Here the row player is trying to maximize their payoff given by the following matrix.

	C_1	C_2	C_3
R_1	5	3	6
R_2	3	2	7
R_3	-1	4	-1

Assuming the row-player goes first, write a linear program to find their optimal strategy.

Use $x_1, x_2, \ldots$ as variables in the LP.

1. What is the objective function of the LP?

2. What are the constraints of the linear program?

2 Runtime Analysis (8 points)

What is the runtime of the following piece of code? Write the recurrence.

```
Function what(n) {
      what(n/2)
      what(n/2)
      k = n
      While (k > 1) {
            for i = 1,2,3,...,k {
                  for j = 1,2,3,...,k/2 {
                        print BLAH
                  }
            }
            k = k/4
      }
      what(n/2)
      what(n/2)
}
```

1. Recurrence Relation

2. Runtime =

3 Count-Min Sketch (10 points)

Here is the state of the count-min sketch data structure after it has processed a stream of items.

Hash function h_1	10	5	3	2	17	1	1	1
Hash function h_2	10	6	2	4	6	4	4	4
Hash function h_3	13	3	3	6	1	4	4	$\cancel{4}$ 6
Hash function h_4	1	2	3	4	5	6	9	10

For an element A, let $trueCount(A)$ denote the total number of occurrences of A in the stream, and let $estimate(A)$ denote the estimate of the number of occurrences of A as per the count-min data structure.

1. What is the length of the stream?

2. What is the largest possible value of $trueCount(A)$ for an element A?

3. What is the smallest possible value of $estimate(A)$ for an element A?

4. For two elements A and B, what is the maximum possible value of $trueCount(A) + trueCount(B)$?

5. What is the largest possible value of $estimate(A) + estimate(B)$?

4 Integrality of Maximum Flow (3 points)

Every network with integer edge capacities has a maximum flow where the amount of flow on each edge is an integer. Briefly justify.

5 MinCut (4 points)

Let $G = (V, E)$ be a network with source s and sink t. Note that G may have more than one minimum s-t cut and more than one maximum s-t flow.

For each of the following, let B be the value of the maximum s-t flow of G. Fill the appropriate circle in each of following cases.

1. Let e be an edge across a minimum s-t cut. Suppose we decrement the capacity of edge e by 1, what is the value of the maximum s-t flow in the resulting network?
   - ○ $B - 1$
   - ○ B
   - ○ Depends on the network.

2. Let e be an edge on one of the maximum s-t flows. Suppose we decrement the capacity of edge e by 1, what is the value of the maximum flow in the new network?
   - ○ $B - 1$
   - ○ B
   - ○ Depends on the network.

6 Dynamic Programming Orders (8 points)

A dynamic programming algorithm has inputs $X[1,\ldots,n]$ and $Y[1,\ldots,n]$, subproblems $E[i,j]$ for all $i,j \in \{1,\ldots,n\}$, and the following recurrence relation,

$$E[i,j] = \min \begin{cases} E[i-1,j-1]+1 & \text{if } X[i] = Y[j-1] \\ E[i-2,\lceil j/2\rceil]+3 & \text{if } X[i] = Y[j] \\ E[i-1,j+1]+1 & \text{if } X[i+1] = Y[j+5] \end{cases}$$

1. Fill in the blanks in the following pseudocode for the DP algorithm. (It is OK for the code to go out of bounds on E. In other words, assume that $E[i,j]$ is well-defined and already computed correctly for i, j in $\{-2,-1,0,n+1,n+2\}$)

   for $\boxed{}$ from 1 to n do {

   for $\boxed{}$ from 1 to n do {

$$E[i,j] = \min \begin{cases} E[i-1,j-1]+1 \quad \text{if } X[i] = Y[j-1] \\ E[i-2,j/2]+3 \text{ if } X[i] = Y[j] \\ E[i-1,j+1]+1 \text{ if } X[i+1] = Y[j+5] \end{cases}$$

   }

   }

2. Suppose our goal is to compute $E[n,n]$ in polynomial time. What is the smallest memory with which you can implement the above algorithm?

   $\boxed{}$

   Describe your implementation in a couple of sentences.

7 NP-completness true/false (10 points)

For each of the following questions, there are four options:

(1) True (T); (2) False (F); (3) True if and only if $\mathbf{P} = \mathbf{NP}$; (4) True if and only if $\mathbf{P} \neq \mathbf{NP}$.

Circle one for each question.

Note: By "reduction" in this exam it is always meant "polynomial-time reduction with one call to the problem being reduced to."

1. There is a reduction from Independent Set to the Longest Increasing Subsequence problem.

○T ○F ○$\mathbf{P} = \mathbf{NP}$ ○$\mathbf{P} \neq \mathbf{NP}$

2. Every problem in P reduces to the 3-SAT problem.

○T ○F ○$\mathbf{P} = \mathbf{NP}$ ○$\mathbf{P} \neq \mathbf{NP}$

3. Every problem in NP reduces to the 3-SAT problem.

○T ○F ○$\mathbf{P} = \mathbf{NP}$ ○$\mathbf{P} \neq \mathbf{NP}$

4. 3-SAT problem reduces to every NP-complete problem.

○T ○F ○$\mathbf{P} = \mathbf{NP}$ ○$\mathbf{P} \neq \mathbf{NP}$

5. The 3-SAT problem reduces to some problem in P.

○T ○F ○$\mathbf{P} = \mathbf{NP}$ ○$\mathbf{P} \neq \mathbf{NP}$

8 True/False. (14 pts)

1. The worst case complexity of the simplex algorithm for linear programming is exponential, but linear programming is in P.

   ◯ True ◯ False

2. For every directed graph, there exists a starting point v, such that calling the $explore(v)$ routine in DFS will visit every node in the graph.

   ◯ True ◯ False

3. For every directed graph and for each SCC in the graph, there exists a starting point v, such that calling the $explore(v)$ routine will visit every node in that SCC of the graph.

   ◯ True ◯ False

4. For a graph G, there is always some MST of G that does not contain the heaviest edge.

   ◯ True ◯ False

5. Suppose u, v are children of the root in a BFS search tree of a connected undirected graph. Then deleting the root will always disconnect u and v in the original graph.

   ◯ True ◯ False

6. Suppose u, v are children of the root in a DFS search tree of a connected undirected graph. Then deleting the root will always disconnect u and v in the original graph.

   ◯ True ◯ False

7. If there is a polynomial time algorithm for 3-SAT then there is a polynomial time algorithm to factor n bit numbers.

◯ True	◯ False

9 Fill in the Blanks (24 points)

When asked for a bound, always give the tightest exact bound possible, not an asymptotic one. Some questions have choices in parentheses after the answer box.

1. Dr. Hurry is an impatient man. He is reading the weights on the edges of a graph G one at a time. After reading just k edges, Dr. Hurry exclaims that an edge e is not part of an MST of G. What is the smallest possible value of k?

2. Let us suppose we execute the Follow the regularized leader (FTRL) algorithm over convex set $[-1, 1]$, with regularizer $R(x) = x^2$. Suppose the cost functions in first three rounds are $f_1(x) = 1 + x$, $f_2(x) = 1 - x$ and $f_3(x) = 1 + x$. Then, the value of x suggested by FTRL algorithm for the fourth step is $= \arg\min_{x \in [-1,1]}($ $)$.

3. Setting the regularizer function to be a constant function $R(x) = 10$ in any FTRL algorithm, we recover the algorithm.

4. If a graph G has some vertex cover of size k, the size of the maximum matching is at most

5. If a graph G has a maximum matching of size k, the size of the minimum vertex cover is at most

6. Suppose we draw a hash function $h : \mathbb{Z}_p \to \mathbb{Z}_p$ from a universal/pairwise independent hash family $\mathcal{H}$, then the probability that $h(100) = h(10)^2 + 5 \mod p$ is at most

7. The greedy algorithm for HornSAT returns the assignment 00011 on a formula Φ with 5 variables. What is the maximum number of satisfying assignments that formula can have?

8. A directed graph has a cycle if and only if every depth-first search on the graph reveals a

(tree/forward/back/cross) edge.

10 Better-Than-Most TSP Tour (10 points)

An instance of TSP consists of n cities and distances $d[\cdot,\cdot]$ between every pair of cities. The distances may not satisfy the triangle inequality. A TSP tour is a path that visits every city exactly once.

There are $(n-1)!$ possible TSP tours in any instance with n cities. Finding the TSP tour that has the smallest total length among all these $(n-1)!$ tours is NP-hard. For distances that don't satisfy the triangle inequality, there are no approximation algorithms for the problem either.

Let us say that a TSP tour is *Better-Than-Most* if its cost is smaller than 99% of the $(n-1)!$ possible TSP tours.

1. Describe an algorithm that given δ in $(0,1)$, runs in polynomial-time in n and $1/\delta$, and outputs some tour which is a *Better-Than-Most* TSP tour with probability $1-\delta$.

   (*Hint*: Given two tours, comparing their costs takes linear time.)

2. What is the runtime of your algorithm in terms of n and δ?

3. Prove that the algorithm outputs a *Better-Than-Most* TSP tour with probability at least $1 - \delta$.

11 Coffee Shops (20 points)

A rectangular city is divided into a grid of $m \times n$ blocks. You would like to setup coffee shops so that for every block in the city, either there is a coffee shop within the block or there is one in a neighboring block. (There are up to 4 neighboring blocks for every block).

It costs r_{ij} to rent space for a coffee shop in block ij.

1. Write an integer linear program to determine which blocks to setup the coffee shops at, so as to minimize the total rental costs.

   (a) What are your variables, and what do they mean?

Variables	What does the variable mean?

   (b) What is the objective function?

   (c) What are the constraints?

Constraint	What does the constraint enforce?

2. Solving the linear program gets you a real valued solution. How would you round the LP solution to obtain an integer solution to the problem? Describe the algorithm in at most two sentences.

3. What is the approximation ratio obtained by your algorithm?

4. Briefly justify the approximation ratio.

12 NP-Completeness Reductions (20 points)

Show that the following problems are NP-complete by providing a polynomial-time reduction. You may assume that the following problems are NP-complete: Rudrata (Hamiltonian) Path, Rudrata (Hamiltonian) Cycle, Vertex Cover, Independent Set and 3-SAT.

1. **Quarter Path**
   Input: Graph $G = (V, E)$, and vertex s
   Solution: There is a path with $n/4$ edges all of whose vertices are distinct

   Proof. Briefly argue that the Quarter Path problem is in NP

   We will now use a reduction to show that the problem is NP-complete. Fill up the details in the proof below..

   We will exhibit a polynomial time reduction from the to the

   Given an instance Φ of the problem we construct an instance Ψ of the problem

   as follows ...

The proof that this is a valid reduction is as follows:

13 Just Repetition (22 points)

Dee is texting Matt using her faulty phone that inserts spurious characters into the message. To cope with these spurious characters, Dee repeats her message twice. We will devise an algorithm for Matt to recover Dee's message from what he receives.

Formally, call a string Y to be a *valid message* if Y consists of some string w repeated twice, i.e., Y is w concatenated with w for some string w.

Matt receives an input string $x[1, \ldots, n]$. Design an algorithm to find the minimum number of character deletions needed to make x into a *valid message*. Your algorithm should take time at most $O(n^3)$.

(*Hint:* use a DP algorithm as a subroutine to solve the problem)

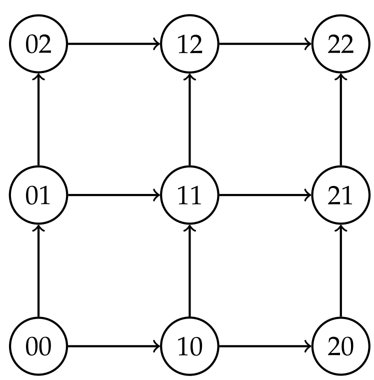
1. Describe the main idea behind the algorithm in three to four sentences.

2. What are the subproblems in the DP?

3. Write the recurrence relation.

14 Multiplicative Weights (12 points)

Consider the following simplified map of Berkeley. Due to traffic, the time it takes to traverse a given path can change each day. Specifically, the length of each edge in the network is a number between $[0, 1]$ that changes each day. The travel time for a path on a given day is the sum of the edges along the path.



For T days, both Max and Vinay drive from node 00 to node 22.

To cope with the unpredictability of traffic, Vinay builds a time machine and travels forward in time to determine the traffic on each edge on every day. Using this information, Vinay picks the path that has the smallest total travel time over T days, and uses the same path each day.

Max wants to use the multiplicative weights update algorithm to pick a path each day. In particular, Max wants to ensure that the difference between his expected total travel time over T days and Vinay's total travel time is at most $T/10000$. Assume that Max finds out the lengths of all the edges in the network, even those he did not drive on, at the end of each day.

1. How many experts should Max use in the multiplicative weights algorithm?

2. What are the experts?

3. Given the weights maintained by the algorithm, how does Max pick a route on any given day?

4. The regret bound for multiplicative weights is as follows:

   Theorem. Assuming that all losses for the n experts are in the range $[0, 4]$, the worst possible regret of the multiplicative weights algorithm run for T steps is

   $$R_T \leq 8\sqrt{T \ln n}$$

   Use the regret bound to show that expected total travel time of Max is not more than $T/10000$ worse than that of Vinay for large enough T.