Final

Name:

SID:

Name and SID of student to your left:

Name and SID of student to your right:

Exam Room:

Rules and Guidelines

- **The exam has 20 pages, is out of 190 points, and will last 170 minutes.**
- 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.
- When there is a box for an answer, **only the work in box provided will be graded.**
- You may use the blank page on the back for scratch work, but it will not be graded. Box numerical final answers.
- Question 1 is true/false on Pre-MT2. Question 2 is true/false on Post-MT2. Question 3 is short answer on Pre-MT2. Question 4 is short answer on Post-MT2. Questions 5 to 9 are longer questions.
- **The problems do 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.
- You may assume that comparison of integers or real numbers, and addition, subtraction, multiplication and division of integers or real or complex numbers, require $O(1)$ time.
- There are warmup questions on the back page of the exam for while you wait.
- Good luck!

This page is deliberately blank. You may use it to report cheating incidents. Otherwise, we will not look at it.

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.**

- ☐ Perla, Monday 9 - 10 am, Dwinelle 243
- ☐ Jenny, Monday 9 - 10 am, Soda 320
- ☐ Sean, Monday 10 - 11 am, Cory 241
- ☐ Yining, Monday 10 - 11 am, Wheeler 222
- ☐ Jerry, Monday 11 am - 12 pm, Etcheverry 3113
- ☐ Jeff, Monday 11 am - 12 pm, Dwinelle 130
- ☐ Peter, Monday 12 - 1 pm, Evans 3
- ☐ David, Monday 12 - 1 pm, Wheeler 108
- ☐ Nate, Monday 12 - 1 pm, Soda 320
- ☐ James, Monday 1 - 2 pm, Dwinelle 182
- ☐ Mudit, Monday 1 - 2 pm, Soda 320
- ☐ Arun, Monday 1 - 2 pm, Barker 110
- ☐ Jierui, Monday 2 - 3 pm, Evans 9
- ☐ Simin, Monday 2 - 3 pm, Evans 70
- ☐ Brandon, Monday 2 - 3 pm, Dwinelle 242
- ☐ Ming, Monday 3 - 4 pm, Cory 289
- ☐ Harley, Monday 3 - 4 pm, Evans 9
- ☐ Aarash, Monday 4 - 5 pm, Dwinelle 79
- ☐ Vinay, Monday 4 - 5 pm, Etcheverry 3119
- ☐ Zheng, Monday 4 - 5 pm, Evans 70
- ☐ Zihao, Monday 5 - 6 pm, Dwinelle 79
- ☐ Max, Monday 5 - 6 pm, Dwinelle 243
- ☐ Matthew, Tuesday 9 - 10 am, Wheeler 108
- ☐ Ajay, Tuesday 9 am - 10 am, Etcheverry 3113
- ☐ Nick, Tuesday 11 am - 12 pm, Etcheverry 3111
- ☐ Sam, Tuesday 12 - 1 pm, Etcheverry 3111
- ☐ Julia, Tuesday 1 - 2 pm, Etcheverry 3119
- ☐ Don't attend Section.

1 True/False - Pre-MT2

(3 points each)

(a) $10000n^2 = O(n^2 \log n)$.

○ True ○ False

(b) e^{cn} is $O(e^n)$ for all $c > 0$.

○ True ○ False

(c) For $T(n) = 16T(n/9) + n^{3/2} \log n$, $T(n) = \Theta(n^{3/2} \log n)$

○ True ○ False

(d) It suffices to choose $1 + \max(\deg(f(x)), \deg(g(x)))$ points if we use Fast Fourier Transform to get the coefficients of $p(x)$ where $p(x)=f(x) \cdot g(x)$.

○ True ○ False

(e) The node with the highest **post**-order number in a depth first search of a directed graph must be in a **source** SCC.

○ True ○ False

(f) The node with the highest **pre**-order number in a depth first search of a directed graph must be in a **sink** SCC.

○ True ○ False

(g) Any connected undirected graph where depth first search does not find a back edge is a tree.

○ True ○ False

(h) Let G be an undirected graph with edge weights w, $f(\cdot)$ be some strictly increasing function, $F(G)$ be the graph G with edge weights w replaced with $f(w)$, and $W(G)$ denote the total weight of a graph G.

(i) Let T be an MST of a graph G. $F(T)$ is an MST of $F(G)$.

○ True ○ False

(ii) Let T_1 and T_2 be spanning trees of G. If $W(T_1) > W(T_2)$ then $W(F(T_1)) > W(F(T_2))$.

○ True ○ False

(iii) Let T be a maximum spanning tree of a graph G. $F(T)$ is a maximum spanning tree of $F(G)$.

○ True ○ False

(i) Recall that in a two person zero sum game with payoff matrix A, the entry $a_{i,j}$ in A is the payoff if the row player plays strategy i and the column player plays strategy j. We say a column j is dominating if for each row i, $a_{i,j} < a_{i,j'}$ for $j' \neq j$. (i.e., for any option the row player picks, the payoff is minimized by picking column j.)

(i) If there is a dominating column, then there is a pure strategy which is optimal for the column player.

○ True ○ False

(ii) If there is a dominating column, there is a pair of pure strategies which are simultaneously optimal for the row and column player.

○ True ○ False

2 True/False - Post-MT2

(3 points each)

(a) If problem A reduces to problem B, then B can be solved in polynomial time **only if** A can be solved in polynomial time.

○ True ○ False

(b) If we can prove an NP-hard problem is in P, then P = NP.

○ True ○ False

(c) If P $\neq$ NP, Vertex Cover can be reduced to Bipartite Matching.

○ True ○ False

(d) Notice that for any Travelling Salesman Instance, we can view the input as a weighted complete graph where edge (u, v) has weight $d(u, v) \geq 0$. The weight of the minimum spanning tree in this graph is a lower bound on the cost of the optimal Travelling Salesman Tour.

○ True ○ False

(e) If for some $a \not\equiv 0 \pmod{N}$, $a^{N-1} \not\equiv 1 \pmod{N}$, then N is not prime.

○ True ○ False

(f) Any two-qubit quantum state can be decomposed into two one-qubit states.

○ True ○ False

(g) In the experts problem with n experts, let A be *any* algorithm which only picks predictions made by an expert that has made no mistakes. A makes $O(\log n)$ mistakes if there is an expert who makes no mistakes.

○ True ○ False

(h) In the weighted majority algorithm, we multiply an expert's weight by $1 - \epsilon$ anytime they make a mistake. To minimize the upper bound on our regret, we should set ϵ to be relatively small if the number of experts is small and the number of mistakes the best expert makes is large.

○ True ○ False

(i) Suppose we modify the weighted majority algorithm so that it multiplies an expert's weight by $1/(1 - \epsilon)$ every time the expert is correct instead of multiplying their weight by $1 - \epsilon$ every time the expert makes a mistake. Then the algorithm achieves the same guarantee.

○ True ○ False

3 Short Answer - Pre-MT2

(4 points each)

(a) Let ω be a primitive nth root of unity for an even number n, and let $S = \{\omega^0, \omega^1, \ldots, \omega^{n-1}\}$. How big is the set $\{x^2 \mid x \in S\}$?

(b) We have an array of n integers A where n is a power of 2. We want to find $f(A) = \max_{j<k}(A[k] - A[j])$, i.e. the maximum of any element minus an element to its left, using a divide and conquer algorithm.

Let L be the left half of A, and R be the right half. Suppose our algorithm has recursively computed $f(L), f(R)$. The algorithm should finish by computing and outputting ________. Write your answer in terms of $f(L), f(R)$ and the elements of L and R.

(c) Let $d = 2^k - 1$ for positive integer k, and let F, G, and H be polynomials of degree at most d satisfying $F(x)/G(x) = H(x)$. Suppose for any 2^k-th root of unity z that $G(z) \neq 0$. Briefly explain how to compute the coefficients of H given the coefficients of F and G in $O(d \log d)$ time.

(d) We have an undirected graph $G = (V, E)$. With probability $0 \leq p_e < 1$, each edge e will independently be deleted from the graph. Given $s, t \in V$, we want to find a path from s to t with the maximum probability of existing in the graph after deletions. In particular, we want to do this by weighting the edges in G and running a shortest path algorithm.

(i) What should be the weight of edge e in terms of the deletion probability p_e?

(ii) Which shortest path algorithm should we run?

○ Dijkstra's ○ Bellman-Ford

(e) We have an undirected weighted graph $G = (V, E)$. For sets of vertices $S, T \subseteq V$, we want to find the shortest path from any vertex in S to any vertex in T. In particular, we would like to do this by adding new vertices and edges to G to get a graph G' and running a shortest path algorithm on G'.

(i) What *new* vertices and edges should we add to G to get G'?

Vertices:

Edges:

(ii) What weight should the new edges have?

(f) What is the average bit length per character in the optimal encoding for the following set of characters and frequencies: $(C, .7), (T, .2), (G, .05), (A, .05)$?

(g) Recall that an increasing subsequence is a sequence $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ where $i_j < i_{j+1}$ and $a_{i_j} < a_{i_{j+1}}$. What is the recurrence for the Longest Increasing Subsequence problem on input $a_1, \ldots, a_n$? Write your answer in terms of subproblems $L(i)$, where $L(i)$ is the length of the longest increasing subsequence ending at i.

(h) Consider the problem of finding the number of paths (with repeated edges allowed) from s to t with 2^k edges in an (unweighted) graph G. In a dynamic program, we define the subproblems $C(u,v,i)$ to be the number of paths from u to v of length 2^i.

(i) Give a recurrence for $C(u,v,i)$.

(ii) Using this idea, give a runtime for computing the number of paths from s to t in terms of the number of vertices, n, the number of edges, m, and k (You may assume that arithmetic operations can be done in $O(1)$ time.)

(i) Given the payoff matrix $\begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}$ for a two person game, what is the best defense strategy for the row player? (Specify as a vector.)

(j) You are given a flow network $G = (V, E)$ with edge capacities $c(\cdot)$ and maximum flow value of F, and a valid flow $f(\cdot)$. What is the value of the minimum s-t cut in the **residual network** in terms of F and the amount of flow, $|f|$, routed by $f(\cdot)$ from s to t?

(k) Consider a linear program $\max c^T x$, s.t. $Ax \leq b, x \geq 0$. For x and y being feasible primal and dual solutions the value of the vector $y^T b - c^T x$ is:

○ ≥ 0 ○ ≤ 0 ○ $= 0$ ○ any real number

4 Short Answer - Post-MT2

(4 points each)

(a) A travelling salesman tour can be viewed as an ordering of the n cities. In a local search approach, what is the number of possible moves from a fixed solution where a move swaps two vertices in the ordering?

(b) Let $|\psi\rangle := \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$, the equal superposition of all n-bit strings. After measuring the first k qubits of $|\psi\rangle$, how many possible outcomes are there if we then measure the whole of $|\psi\rangle$?

(c) How many elements of $\{1, \ldots, 48\}$ have a multiplicative inverse modulo 49?

(d) We are using the weighted majority algorithm for the experts problem. Suppose there are n experts, we multiply an expert's weight by $(1 - \epsilon)$ anytime they are wrong, and we make the wrong decision for the first t days. Give an upper bound on the total weight of all experts at the end of day t.

(e) Recall that with n experts and 1 perfect expert that the algorithm of following the advice of the majority of experts who have not made a mistake ensures that we don't make more than $\log n$ mistakes. Give an improved bound on the number of mistakes if k of the experts are perfect.

5 Reductions

(a) (**8 points**) Consider a weighted directed graph $G = (V, E)$ with integer edge weights. In the minimum cycle cover problem (call this problem **MCC**) we want to find the minimum weight set of simple cycles where every vertex participates in exactly one cycle. Give a reduction from this problem to the problem of finding a perfect matching of minimum weight in a bipartite graph (call this problem **MWPBM**).

(i) Fill in the blank: We take an instance of ____
and create an instance of ____

(ii) Describe the graph created by the reduction:
Vertices:

Edges:

(iii) Given a solution to the reduced instance, describe how to retrieve a solution to the original instance:

(b) (**4 points**) Notice that a travelling salesman tour is a cycle cover in the weighted complete graph on the cities. Does this imply that minimum cycle cover is NP-hard? Briefly justify your answer.

(c) (**8 points**) Given a directed graph G with weighted edges, the Shortest Simple Path problem is to find the shortest simple path between vertices s and t. The graph may have negative edges and/or negative cycles. **Recall that simple paths do not repeat vertices.** Show that Simple Shortest Path is NP-Hard by reducing from Rudrata Path.

(Recall that a Rudrata path is a simple path with arbitrary endpoints that includes all the vertices.)

(i) Fill in the blank: We take an instance of ____
and create an instance of ____

(ii) Describe the graph created by the reduction:
Vertices:

Edges:

(iii) Given a solution to the reduced instance, describe how to retrieve a solution to the original instance:

(d) (**8 points**) A k-bounded spanning tree for an undirected graph G is a spanning tree T such that each vertex has at most k neighbors in T. For integer $k \geq 2$ the **k-Bounded Spanning Tree Problem** is: Given a graph G, does a k-bounded spanning tree exist? Notice that when $k = 2$, this would be the Rudrata Path problem.

Give a reduction from Rudrata Path to 10**-bounded spanning tree**.

(i) Fill in the blank: We take an instance of ____
and create an instance of ____

(ii) Describe the graph created by the reduction:
Vertices:

Edges:

(iii) Given a solution to the reduced instance, describe how to retrieve a solution to the original instance:

6 Universal hash functions.

For this problem, we define a family $\mathcal{H}$ of hash functions from S to T to be *universal* if $\Pr_{h \in \mathcal{H}}[h(x) = h(y)] = \frac{1}{|T|}$.

(a) (**6 points**) Consider a universal hash family, $\mathcal{H}$, of hash functions from $\{0, 1, \ldots, m-1\}$ to $\{0, 1, \ldots, n-1\}$.

(i) Given a subset $S \subset \{0, 1, \ldots m-1\}$, give a reasonable upper bound for $\Pr_{h \in \mathcal{H}}[\exists x, y \in S, h(x) = h(y)]$ in terms of $|S|$ and n.

(ii) To estimate the size of S, we hash all the values and check how many buckets get items. Let B be the number of buckets which are not empty. If $|S| \leq k$, what should n be so that $\Pr_{h \in \mathcal{H}}[B = |S|] \geq 1/2$? (Don't worry about small additive constants)

(b) (**6 points**) For each of these hash function families, state if the family is universal. If so, explain why. If not, for some value of m give an example of two inputs that collide with probability greater than $1/m$.

(i) The family containing $h_{a_1,a_2,a_3}(x_1, x_2) = a_1x_1 + a_2x_2 + a_3x_1 \mod m$ for each $a_1, a_2, a_3 \in \{0, \ldots, m-1\}$, where m is prime and x_1, x_2 are in $\{0, \ldots, m-1\}$.

(ii) The family containing $h_{a_1,a_2}(x_1, x_2, x_3) = a_1x_1 + a_2x_2 + a_1x_3 \mod m$ for each $a_1, a_2 \in \{0, 1, \ldots m-1\}$, where m is prime and x_1, x_2, x_3 are in $\{0, 1, \ldots m-1\}$. (Note that the third term in the definition of h has changed).

7 Approximately Finding the Median of a Stream

We want to design a space-efficient algorithm that scans a stream of m integers between 1 and n and outputs the median of the stream when queried. For simplicity, you may assume that m is odd in all parts of the problem. **No proof of correctness is required for any part.**

(a) (**4 points**) Suppose we have an algorithm A which exactly solves the problem, and the stream S has been scanned by A. Show how to retrieve a sorted copy of S from A by only appending new numbers between 1 and n to the input of A and repeatedly querying A for the median. For simplicity, you may assume you already know the length of S.

You will now design a *deterministic* algorithm that outputs an *approximate* answer.

(b) (**4 points**) Consider the simpler problem of outputting **Yes** if the median is a value k (fixed ahead of time), and **No** otherwise. Give a streaming algorithm which does this using at most $O(\log m)$ bits.

(c) (**4 points**) Give an algorithm which uses $O(\log m \log n)$ bits which when queried outputs a number which is at least the median and at most twice the median.

8 Not quite infallible experts.

We are solving the experts problem with n experts, and we know there is a true expert who will make **strictly fewer** than c mistakes for some constant c.

Suppose we run the majority algorithm, but only remove an expert from the set of trusted experts once they have made c mistakes.

(a) (**4 points**) Let ϕ_i be c minus the number of mistakes made by expert i, or 0 if expert i has made c or more mistakes (i.e, $\phi_i \geq 0$ always). Let $\phi = \sum_{i=1}^{n} \phi_i$. Any time this algorithm makes a mistake, by at least what multiplicative factor does ϕ decrease?

(Hint: If the number of trusted experts remaining is t, what's an upper bound on ϕ?)

(b) (**4 points**) Using your answer to the previous part, give an upper bound on the number of mistakes made by this algorithm. (Hint: Use the fact that $(1 - 1/b)^a \leq e^{-a/b}$).

9 Faster Longest Increasing Subsequence.

For input $a_1, \ldots, a_n$ an increasing subsequence is a sequence $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ where $i_j < i_{j+1}$ and $a_{i_j} < a_{i_{j+1}}$. The Longest Increasing Subsequence (LIS) problem is to find an increasing subsequence of maximum length. For the sake of convenience, we assume that all the a_i's are distinct.

A solution to the LIS problem can be found by playing the patience game on a sequence $a_1, \ldots, a_n$ as follows.

1. Place the first element of the sequence in a pile.
2. For each subsequent element a_j, place it on the leftmost pile for which a_j is less than the top card on the pile. If no such pile exists, a_j goes in a new pile to the right of all other piles.
3. Report the number of piles as the length of the longest increasing subsequence.

For example, if we run the algorithm on the sequence $S = 7, 4, 6, 8, 1$, we obtain piles $(7, 4, 1), (6), (8)$.

(a) (**3 points**) **Briefly** describe an **efficient** implementation of this algorithm and state its runtime.

For the subsequent parts, we consider the following augmentation to the algorithm. We add a backpointer from a_i to the top element of the pile to the left *at the time that a_i is placed.* No backpointers are recorded for elements in the first pile.

For example, if we run the algorithm on the sequence $S = 7, 4, 6, 8, 1$ with $(7, 4, 1), (6), (8)$, the backpointer from 8 goes to 6, and the backpointer from 6 goes to 4. Notice that following the backpointers from 8, gives the sequence 8,6,4 whose reverse corresponds to an increasing subsequence of S.

(b) (**4 points**) Prove that the sequence of backpointers form a decreasing sequence and that its reverse forms an increasing sequence in $a_1, \ldots, a_n$.

(c) (**5 points**) Prove that for any element a_j in the ℓ-th pile from the left, the length of the longest increasing subsequence ending at a_j is ℓ.

- **Warmup Celebrity Lookalike Question:** (**0 points**) Which celebrity does head TA Vinay resemble more?

  ○ Bruno Mars ○ Justin Timberlake (during the NSYNC days)

- **Warmup Celebrity Lookalike Question:** (**0 points**) Which celebrity does TA Nick resemble more?

  ○ John Oliver ○ Daniel Radcliffe

- **Warmup Celebrity Lookalike Question:** (**0 points**) Which celebrity does Professor Chiesa resemble more?

  ○ Elon Musk ○ Alessandro Chiesa

- **Warmup Etymology Question:** (**0 points**) What eating utensil is 170 TA Nick named after?

  ○ Spoon ○ Fork ○ Knife ○ Chopsticks

- **Warmup Piazza Etiquette Question:** (**0 points**) If your Piazza question hasn't been answered yet, what should you do?

  ○ Be patient, Arun will answer it eventually (i.e., within the next ten minutes)

  ○ Add a +1 to encourage Mudit to answer it

- **Warmup English Question:** (**0 points**) Recall that k-CLIQUE is NP-complete. How do you pronounce 'clique'?

  ○ cleek ○ click