University of New Mexico
Department of Computer Science

Final Examination

CS 561 Data Structures and Algorithms
Fall, 2024

Name:
Email:

Directions:

- This exam lasts 2 hours. It is closed book and notes, and no electronic devices are permitted. However, you are allowed to use 2 pages of hand-written "cheat sheets"
- *Show your work!* You will not get full credit, if we cannot figure out how you arrived at your answer.
- Write your solution in the space provided for the corresponding problem.

Question	Points	Score	Grader
1	20		
2	20		
3	20		
4	20		
5	20		
Total	100		

1. **Short Answer (4 points each)**

   Answer the following using *simplest possible* Θ *notation*.

   (a) You toss n^2 balls independently and uniformly at random into n bins. What is the expected number of pairs of balls that wind up in the same bin?

   (b) You toss n balls into n^4 bins independently and uniformly at random. Use a union bound to bound the probability (within tight big-O) that any pair of balls fall into the same bin.

   (c) Solution to the recurrence $T(n) = 3T(n/4) + n$?

   (d) Solution to the recurrence: $f(n) = 2f(n-1) - f(n-2) + 1$. Answer in big-O notation here.

(e) A stack has two operations: Push, and a PopGreaterThan(x) operation which repeatedly pops the top item off the stack until either the stack is empty or the top item on the stack has value less than x. Over a total of n operations on an initially empty stack, what is the amortized cost per operation?

2. **MSTs and Gradient Descent**

(a) (10 points) Let T be a minimum spanning tree in some graph $G = (V, E)$. Prove that any edge (u, v) in T is a light edge crossing some cut.

(Remember that a cut is a partition of the nodes in V into two disjoint sets. You will get no points on this problem if you don't specify these two sets for which (u, v) is a light crossing edge. Hint: Use both T and (u, v) in defining your cut.)

(b) (10 points) You have a set, D of data points that each have a width and height value, where all width and height values are normalized to be between 0 and 1. You want to find a point, $x = (x.w, x.h)$ that minimizes squared distances to all data points, i.e. you want to minimize the function $f(x) = \sum_{d \in D} (x.w - d.w)^2 + (x.h - d.h)^2$. You plan to use regular gradient descent to find x.

i. (5 points) Describe the convex search space κ and the diameter D of κ.

ii. (5 points) Describe the gradient function vector, $\nabla f(x)$, by giving the two functions $\frac{\mathrm{d}f}{\mathrm{d}w}$ and $\frac{\mathrm{d}f}{\mathrm{d}h}$ in the gradient vector.

3. **NP-Completeness**

You are planning a dinner party for a set, S, of people, some of whom are enemies. In particular, you have a set of pairs of people, T who are enemies. You also have k tables, each of which can seat any number of people. Your want to minimize the total number of pairs of people who are enemies that are assigned to the same table. In particular, you are given a number, j giving the maximum number of such pairs that can be seated together.

Thus, in the *DINNER-PARTY* problem, you are given a set S; a set T of pairs of elements in S; and integers k and j. You must return TRUE iff there is a way to assign each person in S to one of the k tables such that the number of pairs in T that are seated at the same table is no more than j.

For example, assume the input is: $S = \{Alice, Bob, Carol, Dan\}$; $T = \{(Alice, Bob), (Alice, Carol), (Alice, Dan), (Bob, Carol)\}$; $k = 2$; and $j = 1$. Then you should return TRUE since you can seat $Alice$ at one table, and $Bob, Carol, Dan$ at the other table, and there is only one enemy pair, $(Bob, Carol)$, seated together.

On the other hand if j is changed to 0 above, then the answer becomes FALSE, since there is no way to sit everyone at 2 tables such that *no* pairs in T are seated together.

(a) (16 points) Prove that DINNER-PARTY is NP-Hard by reduction from one of the following: 3-COLORABLE, CLIQUE, INDEPENDENT SET, VERTEX-COVER, 3-SAT, or HAMILTONIAN-CYCLE.

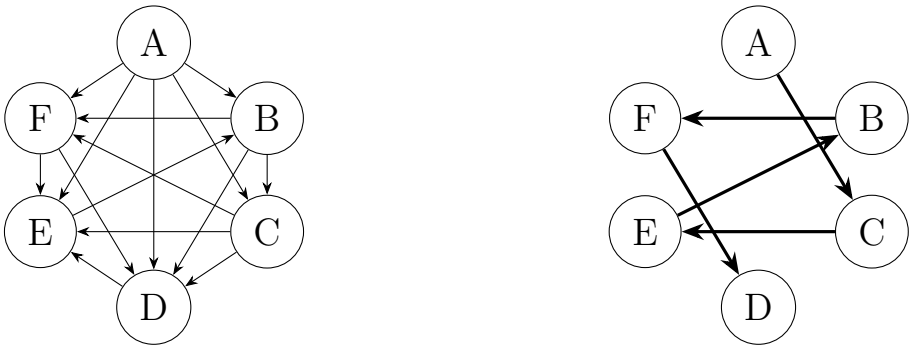
(b) (4 points) Prove that DINNER-PARTY is NP-COMPLETE by showing that it is also in NP.

4. **Induction and Recursion**

A *tournament graph* is a directed graph, $G = (V, E)$ where each pair of nodes has exactly one directed edge between them. I.e., for all $u, v \in V$, either $u \to v$ or $v \to u$ exists in E.

A *Hamiltonian path* in a directed graph, is a directed path that visits each vertex exactly once. In general, determining whether an arbitrary graph has a Hamiltonian path is NP-Hard; but, here you'll be showing that every tournament graph has a Hamiltonian path.

Below is an example tournament graph on the left, with a Hamiltonian path through it on the right.



- (20 points) Prove by induction that every tournament graph has a Hamiltonian path.

HINT: Prove this by induction on n. In your IS, let v be any node, and let V_{in} be the nodes with edges pointing into v; and V_{out} be the nodes to which v points. If you can apply the IH to the sub-graph over V_{in} and the sub-graph over V_{out}, can you then piece together everything to create a Hamiltonian path that includes v? A correct inductive proof also gives a recursive algorithm.

4. **Induction and Recursion, continued.**

5. **Cake Cutting** After the holidays, you buy a large cake wholesale, in order to cut it into smaller cakes to resell. The original cake is a large rectangle that has width m inches and height n inches. You want to cut it into smaller rectangles of various integer dimensions, so as to maximize the sum of the prices of all these rectangles. You have a lookup array that tells you the resale price, $P[x, y]$, of any x-inch by y-inch rectangle. The resale prices are unusual, depending on aesthetics, customer demand, etc., so don't make any assumptions about them.

   You can make horizontal or vertical cuts across any rectangle with your knife, *but you must cut all the way through the rectangle.*

   (20 points) Given integers m and n, and an array P, describe a dynamic program to compute the maximum resell value you can obtain if you subdivide the original cake optimally. Be sure to include the following:

   (a) An English description of the subproblems (e.g. "minimum edit distance of first i characters of string A and first j characters of string B").

   (b) A mathematical description of the recurrence (e.g. "Base case(s): e(0,i) = i, e(j,0) = j; Recurrence e(i,j) = min(e(i,j-1), e(i-1,j), e(i-1,j-1) + 1 - I(A[i]=B[j])").

   (c) How to compute the final answer using the recurrence (e.g. "Return e(m,n)").

   (d) How to solve the subproblems and analysis of your runtime (e.g. "Fill in a table left to right, top down. Runtime is $O(nm)$")

5. **Cake Cutting, continued.**