University of New Mexico
Department of Computer Science

Final Examination

CS 561 Data Structures and Algorithms
Fall, 2020

Name:
Email:

- This exam lasts 4 hours. It is open book, notes and Internet. However, you are not allowed to discuss problems with any person.
- PLEASE: Start each main problem (i.e. Problems 1-5) at the top of a new page!
- Give yourself extra time to properly upload your solutions. Late exams may be penalized.
- *Show your work!* You will not get full credit if we cannot figure out how you arrived at your answer.

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

1. **Short Answer (2 points each)**

   Answer the following questions using the *simplest possible* Θ notation. Assume as usual, that $f(n)$ is $\Theta(1)$ for constant values of n.

   (a) Solution to the recurrence $T(n) = T(n-1) + n$?

   (b) Expected time to search for a key in a skip list storing n items?

   (c) Solution to the recurrence $T(n) = 2T(n/4) + n^2$?

   (d) Solution to the recurrence: $f(n) = 3f(n-1) - 2f(n-2)$.

   (e) In a union-find implementation with union by rank and path-compression, what is the worst case runtime of a single call to FIND-SET?

For a gift exchange, n friends write down their names on a piece of paper and then place these names in a hat. Each friend then draws a name randomly from the hat. Answer the following questions using regular (non-asymptotic) notation.

(f) Consider a single friend. What is the probability that they get back their own name?

(g) What is the expected number of friends who get back their own name?

(h) Use Markov's inequality to get an upper bound on the probability that at least $n/2$ of your friends draw their own name.

(i) Assume n is divisible by 3, and that $2n/3$ friends are known to want fruitcake. Unfortunately, only the remaining $n/3$ friends are able to give fruitcake (due to the great 2020 holiday fruitcake shortage). A person with a fruitcake gifts it to the name they draw iff the name drawn is known to want a fruitcake. What is the expected number of fruitcakes given?

(j) Assume that this gift exchange is repeated 2 times. Use union bounds to bound the probability that *any* friend picks their name both times.

2. **MAX-COLORING** In the MAX-COLORING problem, you are given a graph G and an integer k. You must color the nodes of G with at most k edges in such a way that maximize the number of *satisfied* edges, where an edge is satisfied if its endpoints are colored differently.

3. **Spanning Trees and Cycles**

(a) (10 points) You are given an undirected, connected graph $G = (V, E)$ with positive weights on all the edges. You want to find a spanning tree of G with *maximum* weight, i.e. the sum of the weight of all edges in the spanning tree is maximized over all spanning trees of G. Give an efficient algorithm to solve this problem. Give the runtime of your algorithm as a function of $n = |V|$ and $m = |E|$.

(b) (10 points) Again you are given a undirected, connected graph $G = (V, E)$. Call a subset of edges, F, a *cycle cover* if every cycle in G contains at least one edge in F. In other words, removing the edges of F from G results in an acyclic graph. You want to find a cycle cover, F, of G with *minimum* weight, i.e. the sum of the weight of all edges in F is minimized over all cycle covers. Give an efficient algorithm to solve this and give the runtime of your algorithm as a function of $n = |V|$ and $m = |E|$.

4. **Node Coloring**

Consider a collection of n nodes numbered 1 to n. For all i, $1 \leq i < n$, nodes i and $i + 1$ are connected by an edge with weight $w_{i,i+1} \in \mathcal{R}$, i.e. this weight can be any real number (not just positive).

You want to color each node red or blue. If a pair $(i, i + 1)$ of neighboring nodes are colored *the same*, the cost associated with the edge connecting those nodes is $w_{i,i+1}$; if the pair are colored differently, the cost of the edge is 0. The total cost of a coloring is the sum of the costs of all edges.

(a) (6 points) Describe a dynamic program to output the minimum cost of any coloring. Hint: Let $c(i, r)$ be the minimum cost of coloring nodes 1 through i when node i is colored red. Let $c(i, b)$ be the minimum cost of coloring nodes 1 through i when node i is colored blue. What is the runtime of your algorithm

(b) (7 points) Now you are constrained so that the number of nodes colored red is no more than $n/2$. Describe and analyze a dynamic program to output the minimum cost of any such coloring. Hint: Add another argument to your recurrence relation.

(c) (7 points) Now you will color nodes of a binary tree either red or blue. You are given a tree $T = (V, E)$, where each edge $(u, v) \in E$ has a edge weight, $w((u, v))$ that is a real number. If both endpoints of the edge are colored the same, the cost associated with that edge is $w(e)$, otherwise it is 0. The total cost of a coloring is the sum of the costs of all edges in the tree. There are no constraints on the number of red nodes. Describe and analyze a dynamic program to output the minimum cost of any coloring for T.
Hint: For a given node v, let $c(v, b)$ ($c(v, r)$) be the minimum cost coloring of the subtree rooted at v if v is colored blue (red). It may help to define to define the following: for a node $v \in V$, let $children(v)$ return the set of nodes that are children of v.

5. **Induction and NP-Hardness**

A c-tree is a tree with each node colored either red, green or silver that obeys the following rules.

- Each red node has two children, exactly one of which is green.
- Each green node has exactly one child, which is not green
- Silver nodes have no children.

In the following, let R, G, and S respectively denote the number of red, green, and silver nodes, and n be the total number of nodes.

(a) (10 points) Prove by induction that in any c-tree with $n \geq 1$, $S = R + 1$.

Show that the next problem is NP-Hard via a reduction from one of the following problems: 3-SAT, SET-COVER, VERTEX-COVER, INDEPENDENT-SET, 3-COLORABLE, HAMILTONIAN-CYCLE, or CLIQUE.

(b) (10 points) **WEIGHTED-ITEM-COVER:** You are given (1) a set S of weighted items; (2) a set T of subsets of items; and (3) a number W. You are asked: can you choose a subset S' of items in S with total weight of items in S' no more than W, such that every subset in T contains at least one item in S'? As an example, let $S = \{a, b, c, d\}$, $w(a) = w(b) = w(c) = 1$ and $w(d) = 2$; $T = \{\{a, b, d\}, \{c, d\}, \{b, d\}, \{a, c\}\}$; and $W = 3$. Then the answer is YES since we can set $S' = \{a, d\}$, which has total weight 3 and also ensures that every set in T contains at least one item from S'.

6. **Repeated Rock, Paper, Scissors**

In Repeated Rock, Paper, Scissors, in each round, you output a *probability distribution* over $\{Rock, Paper, Scissors\}$ and then your opponent chooses a value from $\{Rock, Paper, Scissors\}$. As usual, Rock beats Scissors; Scissors beats Paper; and Paper beats Rock. If you lose your cost is 1, if you win, your cost is -1, otherwise, your cost is 0. Your goal is to minimize your total expected cost over T rounds of this game.

In this game, the best offline cost, OPT, equals the maximum, over all probability distributions, of the expected cost over all T rounds using that probability distribution. For example, if $T = 6$ and your opponent played (R, P, S, R, P, S), then $OPT = 0$, which can be obtained using probability distribution $(1/3, 1/3, 1/3)$. However, if your opponent played (R, R, R, R, R, R), then $OPT = -6$, which can be obtained using the probability distribution $(0, 1, 0)$.

You want to be get an expected total cost that is not too much higher than OPT. You decide to use *online* gradient descent to do this. For round i, you let x_i be a vector of length 3 giving the probabilities, x_i^r, x_i^p, x_i^s, that you use in round i for choosing rock, paper, scissors, respectively.

(a) (6 points) What are the functions $f_i(x_i)$ in the online gradient descent algorithm? Hint: There are 3 different functions possible, based on your opponent's choice in round i.

(b) (2 points) In one sentence, argue that these functions are convex.

(c) (2 points) In one sentence, describe both the convex search space κ and D, the diameter of κ?

(d) (2 points) What is G, the max value of $|\nabla f_i(x)|$? over all i, $1 \leq i \leq T$ and all $x \in \kappa$?

(e) (2 points) Recall that online gradient descent ensures that for any $x^* \in \kappa$,

$$\frac{1}{T}\left(\sum_{i=1}^{T} f_i(x_i) - \sum_{i=1}^{T} f_i(x^*)\right) \leq \frac{GD}{\sqrt{T}}$$

Based on this, give a bound on the expected total cost of your algorithm over all T rounds, as a function of OPT and T only.

(f) (6 points) You now want to incorporate "memory" of what your opponent played in the previous round (assume there is one practice round before the first, so that there is always a previous value for your opponent). Now OPT uses 3 probability distributions, one for each possible value of what was played in the previous round. For example, if your opponent plays (R, P, R, P, R, P), then now OPT = -5, which is obtained by using probability distribution $(0, 0, 1)$ when previous round value is R; and probability distribution $(0, 1, 0)$ when the previous round value is P

Briefly describe how you would change online gradient descent to be competitive with this new OPT. Describe changes to the convex search space κ, and the new f_i functions. Give a bound on the expected total cost of your algorithm now as a function of OPT and T.