University of New Mexico
Department of Computer Science

Final Examination

CS 561 Data Structures and Algorithms
Fall, 2019

Name:
Email:

- This exam lasts 120 minutes. It is closed book and notes, and no electronic devices are permitted. However, you are allowed to use 2 pages of handwritten "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.
- If any question is unclear, ask for clarification.

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* asymptotic notation. Assume as usual, that $f(n)$ is $\Theta(1)$ for constant values of n.

   (a) Expected run-time of randomized quick-sort when sorting a list of n items? *Solution:* $\Theta(n \log n)$

   (b) Solution to the recurrence $T(n) = 2T(n/2) + n$? *Solution:* $\Theta(n \log n)$

   (c) Solution to the recurrence $T(n) = 4T(n/2) + n^2$? *Solution:* $\Theta(n^2 \log n)$ *by Master Method.*

   (d) Solution to the recurrence: $f(n) = 2f(n-1) + 2^n$. *Solution:* $\Theta(n2^n)$ *or* $\theta(c_1 2^n + c_2 n 2^n)$. *Annihilator is* $(L-2)^2$.

   (e) In a union-find implementation with union by rank, but *without* path-compression, what is the amortized cost of FIND-SET? *Solution:* $\Theta(\log n)$

(f) Expected number of nodes in a skip-list storing n data items? *Solution:* $O(n)$

(g) Time to compute the way to parenthesize a sequence of n matrices in order to minimize the number of scalar multiplications? *Solution:* $\Theta(n^3)$

(h) You have a sequence of n lights, each of which is colored either red or green independently with probability 1/2. A contiguous subsequence of lights is said to be monochromatic, if all lights in the subsequence are the same color. What is the expected number of monochromatic contiguous subsequences of length $\log_2 n$? *Solution:* $(n - \log n)(1/n) = \Theta(1)$

(i) For this question only, give a number (not Θ notation). Consider the sequence of lights in the last problem. Using Markov's inequality, what is an upper bound on the probability that there are greater than or equal to $3n/4$ red lights total? *Solution: Let $X =$ number of red. Then $E(X) = n/2$, and $Pr(X \geq 3/2E(X)) \leq 2/3$.*

(j) You throw n balls uniformly and independently into n^5 bins. Using a union bound, what is the probability that there is some subset of 2 balls that fall into the same bin? *Solution:* $\Theta(n^{-3})$

2. **Induction and NP-Hardness**

(a) (10 points) Prove by induction that the number of nodes in a binary tree of height h is at most $2^{h+1} - 1$. Don't forget to include BC, IH and IS. (Recall that the height of a rooted tree is the maximum number of edges in a path from the root to any leaf node). Hint: Do induction on h. For the inductive step, what do you get if you remove the root? *Solution: BC:* $h = 0$. *There can be at most* 1 *node, which is at most* $2^1 - 1 = 1$. *IH: For all* $j < h$, *a binary tree of height* j *has at most* $2^{j+1} - 1$ *nodes. IS: Consider a binary tree of height* h. *Remove the root node to obtain at most* 2 *binary trees of height at most* $h - 1$. *By the IH, the total number of nodes in each tree is at most* $2^h - 1$. *Thus the total number of nodes in both trees is at most* $2^{h+1} - 2$. *Adding the original root node back in, we get that the number of nodes in the original tree is at most* $2^{h+1} - 1$.
GRADING: 2 *points BC,* 2 *points correct IH,* 6 *points inductive step.*

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) **RADIO-TOWERS:** You are given (1) a set S of towers; (2) a set T of subsets of towers; and (3) a set of k radio frequencies. Can you assign exactly one of k possible radio frequencies to each tower in such a way that each tower in each set in T has a unique radio frequency? As an example, let $S = \{a, b, c, d\}$, $T = \{\{a, b, c\}, \{a, d\}, \{a, b, d\}$, and $k = 3$. Then the answer is YES since we can assign tower a frequency 1, tower b frequency 2, and towers c and d frequency 3, thereby ensuring that each tower in each set in T has a unique frequency. *Solution: Reduce from 3-COLORABLE. Nodes become towers, edges become conflicting subsets,* $k = 3$.
GRADING: 2 *points choosing correct problem to reduce from,* 2 *points correct setup and direction of reduction;* 2 *points mapping nodes to towers;* 2 *points mapping edges to conflicting subsets;* 2 *points* $k = 3$

3. **Dynamic Programming and Graphs**

Given a list of n integers, $v_1 \ldots v_n$, the product-sum is the largest sum that can be formed by multiplying adjacent elements in the list. Each element can be multiplied with at most one of its neighbors. For example, given the list $1, 2, 3, 1$, the product sum is $8 = 1 + (2 \cdot 3) + 1$, and given the list $2, 2, 1, 3, 2, 1, 2, 2, 1, 2$ the product sum is $19 = (2 \cdot 2) + 1 + (3 \cdot 2) + 1 + (2 \cdot 2) + 1 + 2$.

(a) (2 points) Compute the product-sum of 1, 4, 3, 2, 3, 4, 2. *Solution:* $29 = 1 + (4 \cdot 3) + 2 + (3 \cdot 4) + 2$.

(b) (8 points) Let $m(j)$ be the product sum for $v_1 \ldots v_j$. Give the recurrence relation for $m(j)$. *Solution:* $m(0) = 0$, $m(1) = v_1$, *For all* $j \geq 2$, $m(j) = \max(m(j-1) + v_j, m(j-2) + v_j * v_{j-1})$

GRADING: 2 points correct base case(s), 2 points for recognizing the smaller subproblems are $m(j-1)$ *and* $m(j-2)$*, 4 remaining points for correct solution.*

(c) (10 points) Let $G = (V, E)$ be a directed graph with edge weights. Let $k \geq 0$ be an integer, and let s and t be vertices of V. Give an algorithm that finds the weight of the maximum weight path from s to t using exactly k edges. If a path of k edges does not exist from s to t, then return $-\infty$. What is the runtime of your algorithm?
Hint: Let $m(v, i)$ be the cost of the maximum weight path from s to v that uses exactly i edges, and use dynamic programming. *Solution: Use dynamic programming based on the following recurrence. For integer* $0 \leq i \leq k$ *and vertex* v*, let* $m(v, i)$ *be the most expensive path from* s *to* v *using* i *edges. Then we have:* $m(s, 0) = 0$*;* $m(v, 0) = -\infty$ *for all* $v \neq s$*; and for* $0 < i \leq k$, $m(v, i) = max_{u \in V, u \neq v, (u,v) \in E} m(u, i-1) + c(u, v)$. *Return* $m(k, t)$. *Runtime is* $O(n^2 k)$.
GRADING: Any correct solution that is $O(n^2 k)$ *gets full credit.* 2 *points correct base case(s);* 2 *points for the correct sub-problems.* 2 *points for correctly adding the edge costs to the cost of the sub-problem;* 2 *points for correct recurrence relation and* 2 *points for correct runtime.*

4. **Number-Guess**

In each round of the Number-Guess problem, you try to guess a value close to an unknown number in the range $[0, 1]$. In round i, you guess a value x_i, and then the actual number v_i is revealed, and your cost for that round is $(x_i - v_i)^2$. You play T rounds, and want to be get a total cost that is not too much higher that the best offline cost, $OPT = \min_x \sum_{i=1}^{T}(x - v_i)^2$. You decide to use *online* gradient descent to do this.

(a) (2 points) What are the functions $f_i(x_i)$ that you are minimizing in the online gradient descent? *Solution:* $f_i(x_i) = (x_i - v_i)^2$

(b) (2 points) What is the gradient of $f_i(x_i)$? *Solution:* $\frac{\partial f_i}{\partial x_i} = 2(x_i - v_i)$

(c) (3 points) In one sentence, describe both the convex search space κ and D, the diameter of κ? *Solution: κ is the line segment* $[0, 1]$*, $D = 1$.*

(d) (3 points) What is G, the max value of $|\nabla f_i(x)|$? over all i, $1 \leq i \leq T$ and all $x \in \kappa$? *Solution:* $G = 2$

(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 total cost of your algorithm over all T rounds, as a function of OPT and T only. *Solution:* $OPT + 2\sqrt{T}$

You now want to define a "hipster"-variant of Number-Guess, where the goal is to choose a number in $[0, 1]$ that is far from the hidden number in each round.[1]

(f) (2 point) Your (unhip) boss suggests, let $OPT = \min_{x\in[0,1]} \; -\sum_{i=1}^{T}(x - v_i)^2$; and let $f_i(x_i) = -(x_i - v_i)^2$, and then use online gradient descent to track OPT. In one short sentence, state why this fails.*Solution:* f_i *is not convex!*

(g) (6 points) Your (hip) barista is really into "underground" means, like the geometric mean. He says to let $OPT = \min_{x\in[0,1]} \; -\left(\prod_{i=1}^{T}(x - v_i)^2\right)^{1/T}$, and then to define f_i functions in order to track some (monotonically increasing) function of this new OPT. What are the f_i functions you should now use? Hint: HW Problem.

Solution: The log *function is concave and its negative is convex. Thus, we can let* $f_i(x_i) = -\log(x_i - v_i)^2$. *Then online gradient descent will track* log *of this new OPT. (This is similar to the trick from the portfolio management problem in the hw.)*
GRADING: 2 *points for the idea of taking the negative of the log.* 4 *points for the correct functions.*

[1] For example, the hidden number is the location of the most-popular radio station on the FM dial, which you want to avoid. (Although real hipsters probably only listen to AM radio:)

5. **Challenge Problem**

In the MAX-COLORING problem, you are given a graph $G = (V, E)$. You must color each node with one of 2 different colors. An edge is *satisfied* if its endpoints are colored differently. Your goal is to find a coloring that **maximizes** the number of satisfied edges. In this problem, you will develop an approximation algorithm for MAX-COLORING, based on a randomized rounding of a linear program.

(a) (10 points) Write an integer program for MAX-COLORING. To do this, create variables x_v for every every vertex, whose value depends on the color of the vertex. Create additional variables z_e for every edge e, whose value is 1 iff e is satisfied. Hint: The z_e constraints are the hard part - think about having **two** constraints for each edge $e = (u, v)$: the first ensures that z_e is 0 when $x_u = x_v = 0$; the second ensures that z_e is 0 when $x_u = x_v = 1$. *Solution: Maximize* $\sum_e z_e$ *subject to* $x_v, z_e \in \{0, 1\}$ *for all vertices* v *and edges* e. *For all edges* $e = (u, v)$ *add the constraints:* $z_e \leq x_u + x_v$ *and* $z_e \leq (1 - x_u) + (1 - x_v)$.
GRADING: 2 *points for the correct constraints on the* x_v *and* z_e *variables;* 2 *points for correct maximization function;* 3 *points for each of the correct* z_e *constraints.*

(b) (2 points) Now consider a relaxation of your integer program to a linear program, where the x_v variables can take on real numbers in the range 0 to 1. Let x_v^* and z_e^* be the settings of the variables in the solution found by the LP. Then, for each vertex v, with probability x_v^* color v with the color 1, otherwise color it 0.
For a fixed edge $e = (u, v)$, what is the probability that e is satisfied as a function of x_u^* and x_v^*? *Solution:* $x_u^*(1 - x_v^*) + (1 - x_u^*)x_v^*$

(c) (8 points) Now let OPT be the optimal solution to MAX-COLORING on G. Give a good lower bound on the expected number of edges satisfied by the rounding. *Solution: For each edge* $e = (u, v)$*, let* Y_e *be a random variable that is* 1 *if the edge is satisfied in the rounding and* 0 *otherwise. Then* $E(Y_e) = x_u^*(1 - x_v^*) + (1 - x_u^*)x_v^*$*. By the LP,* $z_e^* \leq x_u^* + x_v^*$ *and* $z_e \leq (1 - x_u^*) + (1 - x_v^*)$*. We now show that* $E(Y_e) \geq (1/4)z_e^*$ *always.*
Case 1: $x_u^* \geq 1/2$ *and* $x_v^* \geq 1/2$*. Then,* $E(Y_e) \geq 1/2((1 - x_v^*) + (1 - x_u^*)) \geq 1/2(z_e^*)$*.*
Case 2: $x_u^* \leq 1/2$ *and* $x_v^* \leq 1/2$*. Then,* $E(Y_e) \geq 1/2(x_v^* + x_u^*) \geq 1/2(z_e^*)$*.*
Case 3: $x_u^* \leq 1/2$ *and* $x_v^* \geq 1/2$*. Thus,* $E(Y_e) \geq x_u^*(1 - x_v^*) \geq 1/4 \geq (1/4)z_e^*$*.*
Case 4: $x_u^* \geq 1/2$ *and* $x_v^* \leq 1/2$ *is symmetric to Case 3.*
Finally, by linearity of expectation, the expected number of edges satisfied is at least $\sum_e (1/4)z_e^* = (1/4)OPT$*.*
GRADING: 4 *points for getting some bound on* $E(Y_e)$*.* 4 *points for getting that this is at least* $z_e/4$*, and thus that the total expectation is at least* $(1/4)OPT$*.*