CS 170 Efficient Algorithms and Intractable Problems
Spring 2016 Alessandro Chiesa and Umesh Vazirani **Final**

Name:

SID:

GSI and section time:

Write down the names of the students on your left and right as they appear on their SID.

Name of student on your left:

Name of student on your right:

Name of student behind you:

Name of student in front of you:

Instructions: You may consult two handwritten, double-sided sheet of notes. You may not consult other notes, textbooks, etc. Cell phones and other electronic devices are not permitted.

Blank policy: Blanks will receive 0 points. Whether you have a solution or just some progress on a problem, make sure to cross out any extraneous work: we will reward succinct/clear solutions and penalize extra text. (Don't write down a bunch of formulas / attempts in the hopes one is correct; this will receive no partial credit.)

If you finish in the last 15 minutes, please remain seated to avoid distracting your fellow classmates.

There are 7 questions. The last page is page 12. *Please write your name and SID on the top of each sheet, and do not detach the last sheet.*

On questions asking for an algorithm, make sure to respond in the format requested. Write in the space provided. Good luck!

Do not turn this page until an instructor tells you to do so.

1. (40 pts.) Short Answer

(a) **(4 pts.) Primality** Use Fermat's Little Theorem with a base of $a = 3$ to show that 8 is not prime. *Explain your answer in a single sentence.*

(b) **(8 pts.) Hashing**

Given a prime p and $a, b \in \{0, \ldots, p-1\}$, define the function $h_{a,b}(x) = ax + b \bmod p$ where $x \in \{0, \ldots, p-1\}$. Show that $H = \{h_{a,b}\}_{a,b \in \{0,\ldots,p-1\}}$ is a pairwise independent hash function family, i.e., show that for every $x \neq y$ and $c, d \in \{0, \ldots, p-1\}$ it holds that

$$\Pr_{h_{a,b} \leftarrow H}\left[h_{a,b}(x) = c \wedge h_{a,b}(y) = d\right] = \frac{1}{p^2} \ .$$

The notation $h_{a,b} \leftarrow H$ means that $h_{a,b}$ is chosen uniformly at random from H, meaning a and b are chosen independently uniformly at random from $\{0, \ldots, p-1\}$.

(c) **(8 pts.) Zero-sum games** Alice and Bob play a game: each player chooses either X or Y. If both players pick X, Bob pays Alice $1. If both players pick Y, Bob pays Alice $2. Otherwise, Alice pays Bob $1.

(i) Formulate this as a zero-sum game by filling in the table below. Alice is the row player and Bob is the column player. Fill in the values so that Alice is the maximizer and Bob is the minimizer.

		Bob: X	Bob: Y
Alice	X		
	Y		

(ii) What are the optimal strategies and for Alice and Bob respectively? *No justification required.*

Alice's optimal strategy:

probability of playing X: _____

probability of playing Y: _____

Bob's optimal strategy:

probability of playing X: _____

probability of playing Y: _____

(iii) What is the value of the game? *No justification required.*

Value: ☐

(d) **(6 pts.) Complexity** For the following questions, circle the (unique) condition that would make the statement true. *No justification required.*

(i) If B is **NP**-complete, then for any problem $A \in \mathbf{NP}$, there exists a polynomial-time reduction from A to B.

always True True iff $\mathbf{P} = \mathbf{NP}$ True iff $\mathbf{P} \neq \mathbf{NP}$ always False

.

(ii) If B is in **NP**, then for any problem $A \in \mathbf{P}$, there exists a polynomial-time reduction from A to B.

always True True iff $\mathbf{P} = \mathbf{NP}$ True iff $\mathbf{P} \neq \mathbf{NP}$ always False

(iii) Horn SAT is **NP**-complete.

always True True iff $\mathbf{P} = \mathbf{NP}$ True iff $\mathbf{P} \neq \mathbf{NP}$ always False

(e) **(6 pts.) Duality**

For the following linear program, prove that the value of the optimal solution is at most $7/3$ by finding the values of the dual variables y_1 and y_2.

$$\begin{aligned} &\max\ 2x_1 + x_2 \\ &\text{such that } 5x_1 + 2x_2 \leq 5 \\ &\qquad x_1 + x_2 \leq 2 \end{aligned}$$

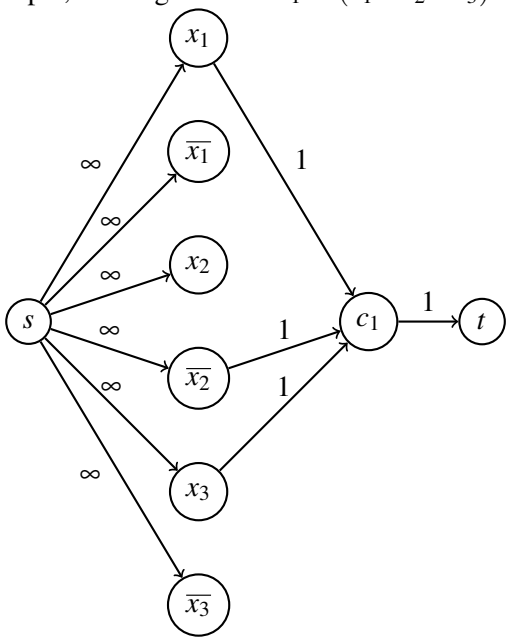
$y_1 =$ $\square$ $\quad y_2 =$ $\square$

(f) **(8 pts.) Bad Reduction** What's wrong with the following reduction from 3SAT to Max Flow? Give a counterexample to show why this reduction does not hold.

Given a 3SAT instance $(c_1, \ldots, c_m)$, we create a Max Flow instance (G, s, t, F) as follows:
For each literal x_i, we create two vertices, x_i and $\overline{x_i}$. For each clause c_j, we create a vertex c_j. We create two more vertices, s and t. Add an edge of infinite capacity from s to each x_i and each $\overline{x_i}$.

Add an edge of capacity 1 from each c_j to t. For each clause c_j, add an edge of capacity 1 from the literals in that clause to the vertex c_j. For example, if $c_5 = (x_2 \vee \overline{x_4} \vee x_9)$, we create edges of capacity 1 from x_2 to c_5, $\overline{x_4}$ to c_5, and x_9 to c_5. Then, G has an $s-t$ flow of F if and only if the 3SAT formula is satisfiable.

For example, the single clause $c_1 = (x_1 \vee \overline{x_2} \vee x_3)$ would yield the following graph:



2. (20 pts.) Cyclic Sorting?

Tanya the TA has a stack of final exams, sorted by score. She wanted to show Prof Vazirani the top score, 105%! Sadly, Charles the clueless TA has split the stack as follows: he picked up the top i exams in the stack, and moved them to the bottom. Help Tanya find the exam with the maximum score again!

Formally, an original sorted array of integers $A[1..n]$ has been cycled: $C = A[i+1..n] \cdot A[1..i]$ for some i such that $0 \leq i < n$, and in this case $\cdot$ is the concatenation operator. You have access to C, but not A or i. Give an efficient algorithm to find the maximum value in the array.

For example, if $A = [-5, -3, 0, 42, 100, 2016]$ and $C = [100, 2016, -5, -3, 0, 42]$, and the maximum value is 2016.

You may assume that all the values in A are distinct.

(a) Give clear pseudocode to solve this problem.

(b) What is the runtime of your algorithm? *No justification required.*

3. (20 pts.) Summer Subletting

You're going to be traveling this summer, and you've listed your apartment on Airbnb. Due to your prime location on Telegraph and Blake and lovely first-story view of the charming streetscape, n guests have expressed interest in renting your apartment. Airbnb is testing out a model where potential guests submit their offer (the total price they're willing to pay for the duration of their stay). Formally, find an efficient algorithm for the following task:

Input: For $1 \leq i \leq n$, (s_i, t_i, p_i) is an offer for a guest who arrives on day s_i, leaves on day t_i, and pays $p_i \geq 0$.

Output: The maximum total value $P^* = \sum_{i \in S^*} p_i$, where $S^* \subseteq \{1, \ldots, n\}$ is the optimal non-overlapping subset of guests. It is OK for a guest to arrive on the same day that the previous guest leaves.

Note: Your runtime should be polynomial in n, the number of guests. If it exceeds our expectations, we will award extra credit.

(a) We'll solve this problem with dynamic programming. First, *clearly* define your subproblems. This consists of defining a function $V(\cdot)$ with some number of arguments, and explaining the meaning of each argument and the value of the function.

(b) Write pseudocode to solve this problem. It should fill in the entries of $V(\cdot)$, and return the maximum total value P^*. Make sure to address the base case(s).

(c) What is the running time of your algorithm? *No justification required.*

4. (20 pts.) The Bold Bluff

You are investing in stocks in the Delphi Exchange. The Delphi Exchange has *only two stocks*, and on each day you either stay with your current stock, or sell all of your holdings and invest all your money in the other stock.

On each day, the price of one stock goes down by 75% (multiplies by 0.25, and this may be a different stock each day), and the other stock doubles. There are 64 oracles in Delphi, who give, on each day, their forecast for the change in stock prices for the next day. You are guaranteed that at least one oracle is always correct.

You start with \$1, and you have 64 oracles. Describe a strategy such that after 28 days, you are guaranteed to have at least \$1000. Justify your answer.

Hint: Don't use the formula from the Multiplicative Weights / Experts lecture.

Main idea:

Analysis:

5. (20 pts.) Rad Reduction You are the captain of a ship, tasked with taking animals across the river. However, not all animals get along – some animals will attack each other. If the ith animal gets attacked, you have to pay the owner a fee of f_i. If the ith animal makes it across the river without getting attacked, you receive a reward of r_i.

In other words, for each animal i, your payoff is as follows:

- 0 if you don't take the animal
- r_i if you take the animal and it doesn't get attacked
- $-f_i$ if you take the animal and it gets attacked

You want to know if there's a subset of animals you can take such that your net profit (the sum of the rewards you get minus the sum of the fees you pay) is at least k.

Input: $[f_1, \ldots, f_n]$; $[r_1, \ldots, r_n]$; k; $E = \{(i, j) : \text{animal } i \text{ attacks animal } j\}$.

Prove that this problem is NP-complete. You may start from the fact that any of the following problems are NP-complete: 3-SAT, circuit-SAT, clique, independent set, TSP, Rudrata path.

Main idea:

Reduction:

Remark: instead of $f_i = \infty$ one could set f_i to any non-negative value.

Proof:

6. (20 pts.) Max 2SAT We saw in class that there is an efficient algorithm to solve 2SAT. Here we consider the Max 2SAT problem: given a set of clauses C such that each clause has at most 2 literals, and integer k, is there a truth assignment that satisfies at least k clauses? We will prove that Max 2SAT is NP-complete by reducing from 3SAT.

(a) Show that the 2SAT formula $(x_1) \wedge (x_2) \wedge (x_3) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_3} \vee \overline{x_1})$ has 3 clauses satisfied if x_1, x_2, and x_3 are either all true or all false, and 4 clauses satisfied otherwise.

(b) Show that the 2SAT formula $(\overline{z}) \wedge (x_1 \vee z) \wedge (x_2 \vee z) \wedge (x_3 \vee z)$ has 4 clauses satisfied if x_1, x_2, and x_3 are all true, and 3 clauses satisfied otherwise.

(c) Show that Max 2SAT is NP-complete.

7. (30 pts.) Fruit Fanatic

(a) You are managing Fred's Fantastic Fruits delivery service. Your job is to choose which fruits to take to the market in your truck. You have n crates of fruit, each of which contains a different type of fruit. The ith crate holds c_i fruits, each of which sells for p_i at the market. Every fruit weighs 1 pound. Your truck can hold T pounds of fruit in total. You are allowed to open all the crates and take any combination of fruits you like, as long as the total weight of those fruits is no more than T. For the remainder of this problem, you may assume that T is an integer and that $c_i \leq T$ for all i.

(i) Describe a greedy strategy to determine the optimal combination of fruits to take (no pseudocode necessary).

(ii) Prove your algorithm is correct using an exchange argument.

(b) You return home from the market dejected, as the fruits didn't survive the long drive to the market in the back of your truck. You realize that the only way to transport them safely is to keep them in their crates. This means that you can either take the ith crate or not – you don't get to take individual fruit out of the crate. However, you realize that this is exactly the knapsack (without replacement) problem!

(i) Reduce this problem to the knapsack problem by defining v_i (the value of the ith knapsack item), w_i (the weight of the ith knapsack item), and W (the total weight of the knapsack) in terms of the variables c_i, p_i, and T from the previous problem.

$$v_i =$$
$$w_i =$$
$$W =$$

(ii) Since you can't solve knapsack exactly, describe an approximation scheme for knapsack that yields at least half the optimal value. Prove that your algorithm gives the desired approximation. You may assume that W and w_i are all integers.
Hint: Use your greedy algorithm from part (a) as inspiration.

Extra page

Use this page for scratch work, or for more space to continue your solution to a problem. If you want this page to be graded, please refer the graders here from the original problem page.