CS 170 Homework 13

Due **5/3/2023, at 10:00 pm (grace period until 11:59pm)**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write "none".

2 Boba Shops

A rectangular city is divided into a grid of $m \times n$ blocks. You would like to set up boba shops so that for every block in the city, either there is a boba shop within the block or there is one in a neighboring block (assume there are up to 4 neighboring blocks for every block). It costs r_{ij} to rent space for a boba shop in block ij.

Write an integer linear program to determine on which blocks to set up the boba shops, so as to minimize the total rental costs.

(a) What are your variables, and what do they mean?

(b) What is the objective function? Briefly justify.

(c) What are the constraints? Briefly justify.

(d) Solving the non-integer version of the linear program yields a real-valued solution. How would you round the LP solution to obtain an integer solution to the problem? Describe the algorithm in at most two sentences.

(e) What is the approximation ratio of your algorithm in part (d)? Briefly justify.

3 Better-Than-Most TSP Tour

An instance of TSP consists of n cities and distances $d[\cdot, \cdot]$ between every pair of cities. The distances may not satisfy the triangle inequality. A TSP tour is a walk that visits every city exactly once and returns to the first visited city.

There are $(n-1)!$ possible TSP tours in any instance with n cities. Finding the TSP tour that has the smallest total length among all these $(n-1)!$ tours is NP-Hard. For distances that don't satisfy the triangle inequality, there are no approximation algorithms for the problem either.

Let us try to approximate TSP from another perspective. We define a TSP tour to be *Better-Than-Most* if its cost is smaller than 99% of the $(n-1)!$ possible TSP tours. Given a constant $\delta \in (0, 1)$, describe an algorithm that outputs a tour that is *Better-Than-Most* with probability $1 - \delta$. Your algorithm should run in polynomial time with respect to n and $1/\delta$. Please provide a 3-part solution.

Hint: Given two tours, comparing their costs takes linear time.

4 Estimating Votes

Suppose we have a stream of votes of form (Id, Yes) or (Id, No) which has a person's Id (that is unique to them) and whether they voted Yes/No. We would like to estimate the fraction of Yes votes. Unfortunately, many people have voted multiple times. People who voted multiple times voted for the same option each time.

The *Distinct Elements* algorithm takes a stream as input, and outputs $\tilde{n} \in [(1-\epsilon)n, (1+\epsilon)n]$ with probability $1-\delta$, where n the number of distinct elements seen in the stream, using small memory. Note that ϵ, δ are constants. Let $S(n, \epsilon, \delta)$ be the space complexity that *Distinct Elements* uses in terms of n, ϵ, δ.

Using the *Distinct Elements* algorithm as a black box, provide an algorithm for estimating the fraction of "Yes" votes within a factor of, say, $(1+3\epsilon)$ with probability at least $1-2\delta$. Justify the correctness of your algorithm and state its space complexity in terms of S.

Hint: assume $\epsilon \leq 1/3$. Then, we have $\frac{1+\epsilon}{1-\epsilon} \leq 1+3\epsilon$.

5 (OPTIONAL) Approximate Median

Let $S = (x_1, x_2, \ldots, x_m)$ denote a stream of m elements. For simplicity, assume that the elements in the stream are unique. Define the *position* of an element to be

$$\text{pos}(x) = |\{y \in S | y \leq x\}|.$$

The ϵ-approximate median is then defined to be an element x such that:

$$\frac{m}{2} - \epsilon m \leq \text{pos}(x) \leq \frac{m}{2} + \epsilon m. \tag{1}$$

Now, provide an algorithm that returns a ϵ-approximate median with high probability. In particular, your algorithm should take ϵ and the failure probability δ as parameters, and return a result that is ϵ-approximate with probability at least $1-\delta$. Please provide a 3-part solution. Note that the optimal algorithm can solve the problem in space independent of the size of the stream.

Hint: Try to provide a sampling based algorithm and argue that less than $1/2$ of the samples will be in the $(0, 1/2-\epsilon)$ percentile using a Hoeffding bound. Similarly, show that less than $1/2$ of the samples will be in the $(1/2+\epsilon, 1)$ percentile.

Hint: The following one-sided version Hoeffding bound might be useful for this (and other) problems: if $X_1, \ldots, X_t$ are i.i.d Bernoulli(p) with $\mathbb{E}(X_i) = p$, we have that:

$$\mathbb{P}\left(\frac{1}{t}\sum_{i=1}^{t} X_i - p \geq \epsilon\right) \leq \exp\left(-2\epsilon^2 t\right).$$

6 (OPTIONAL) Reservoir

(a) Design an algorithm that takes in a stream $z_1, \ldots, z_M$ of M integers in $[n]$ and at any time t can output a uniformly random element in $z_1, \ldots, z_t$. Your algorithm may use at

most polynomial in $\log n$ and $\log M$ space. Prove the correctness and analyze the space complexity of your algorithm. Your algorithm may only take a single pass of the stream. *Hint:* $\frac{1}{t} = 1 \cdot \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdots \frac{t-1}{t}$.

(b) For a stream $S = z_1, \ldots, z_{2n}$ of $2n$ integers in $[n]$, we call $j \in [n]$ a *duplicate element* if it occurs more than once.

Prove that S must contain a duplicative element, and design an algorithm that takes in S as input and with probability at least $1 - \frac{1}{n}$ outputs a duplicative element. Your algorithm may use at most polynomial in $\log n$ space. Prove the correctness and analyze the space complexity of your algorithm. Your algorithm may only take a single pass of the stream.

Hint: Use $\log n$ copies of the algorithm from part a to keep track of a random subset of the elements seen so far. For proof of correctness, note that there are at most n indices t such that $z_t \neq z_{t'}$ for any $t' > t$, i.e. element z_t never occurs in the stream after index t.

7 (OPTIONAL) Comparing Traffic

Alice and Bob want to estimate the following: *Among all vehicles that pass through Hearst Ave and Bancroft Ave on a given day, what fraction of them go through both the streets?*

Alice will observe traffic on Hearst Ave all day, while Bob will observe traffic on Bancroft Ave all day. The two of them will take notes on small pieces of paper. At the end of the day, they would like to compare their notes and estimate the fraction of vehicles in common between the two streets.

Formally, Alice observes a stream $A = \{a_1, \ldots, a_n\}$ and Bob observes a stream $B = \{b_1, \ldots, b_m\}$ where each $a_i, b_j \in \{1, \ldots, N\}$. The goal is to estimate $\frac{|A \cap B|}{|A \cup B|}$.

1. At first, Alice and Bob decide to each store a small random sample of vehicles $S \subset A$ and $T \subset B$ (using say reservoir sampling). At the end of the day, Alice and Bob would compute $\frac{|S \cap T|}{|S \cup T|}$. Briefly argue why this algorithm fails.
2. Using the hashing idea from the streaming algorithm for computing distinct elements, devise an algorithm that Alice and Bob can use.