Practice Quiz 1

- The following practice quiz is a compilation of relevant problems from previous semesters.
- This practice quiz may not be taken as a strict gauge for the difficulty level of the actual quiz.
- The quiz contains multiple problems, several with multiple parts, for a total of 120 points, which should take 120 minutes.
- **This practice quiz is not meant to exactly reflect the difficulty level of the actual quiz, but rather provide a collection of representative problems.**
- Write your solutions in the space provided. If you run out of space, continue on a scratch page and make a notation.
- Do not waste time deriving facts that we have studied. Just cite results from class.
- When we ask you to *give an algorithm* in this quiz, describe your algorithm in English or pseudocode, and provide a short argument for correctness and running time. You do not need to provide a diagram or example unless it helps make your explanation clearer.
- Do not spend too much time on any one problem.
- **Good Luck!**

Problem-1: [10 points] **True or False**

Please circle **T** or **F** for the following. *No justification is needed (nor will be considered).*

(a) [2 points] **T** **F**

In amortized analysis, we require that the real cost of executing any given operation is always upper bounded by the amortized cost of that operation.

(b) [2 points] **T** **F**

Given any two polynomials P and Q, and *any* two real numbers a and b, the discrete Fourier transform of the polynomial $a \cdot P + b \cdot Q$ is equal to $a \cdot P^* + b \cdot Q^*$, where P^* and Q^* are the discrete Fourier transforms of the polynomials P and Q, respectively.

(c) [2 points] **T** **F**

Given an *unsorted* array of n distinct integers, we can output the $n/4$ smallest elements in $O(n)$ time.

(d) [2 points] **T** **F**

Consider a family of hash functions $\mathcal{H} = \{h_1, h_2\}$ that maps a universe $\{a, b, c\}$ to $\{0, 1\}$, where $h_1(a) = h_1(b) = 0$, $h_1(c) = 1$ and $h_2(a) = h_2(b) = 1$, $h_2(c) = 0$. This family $\mathcal{H}$ gives rise to a universal hash function family.

(e) [2 points] **T** **F**

A UNION-FIND data structure of n elements is initially arranged as a binary tree. If the data structure is implemented with path compression, it can be reduced to a height-one tree with $O(\log n)$ calls to FIND-SET. (A height-one tree has all elements directly connected to the root.)

Problem-2: [20 points] **Crossword Contest**

Every day you and your roommate stage a contest to see who can complete the New York Times crossword puzzle faster. The loser pays the winner one dollar. You begin the contest on day 1, and you and your roommate are equally matched, so the probability p that you win is 0.5 for each day.

(a) [10 points] You are happy with this arrangement so long as you don't risk too much money. Show that, for any $t \geq 1$, the probability that the net amount of dollars you lose after t days will be larger than $2\sqrt{t}$ is at most $\frac{1}{8}$.

(b) [10 points] You take the summer off to study for your GREs, and when you restart in the Fall, you are much better than your roommate, with a probability p that you win of 0.75 each day.

You are now looking forward to winning some money. Let W_t be a lower bound on the net amount of money that you will win with probability of at least $1 - e^{-1}$. What is the best (lower bound) estimate on W_t that you can come up with?

Problem-3: [15 points] **Line World**

Ben has a list of the locations x_1, x_2, . . ., x_n of the n cities of Line World, which all lie along the positive x-axis, alphabetized by city name. Ben wants to place $p-1$ post offices in cities to provide equal service to the cities of the world. Thus, Ben decides to place post offices so that they partition the cities into p groups, each of size n/p. For the sake of this problem and Ben's sanity, we assume that this is possible without rounding error.

(a) [5 points] Suppose Ben only has the resources to build 3 post offices (ie. $p = 4$). Which cities should Ben choose? Design an $O(n)$ algorithm to select cities.

(b) [10 points] Now suppose Ben can instead build $p-1$ post offices. Design an algorithm to figure out the location of $p-1$ post offices. For full credit, the algorithm should run in $O(n \log p)$ time.

Problem-4: [30 points] **Antenna Repair**

You are one of the first colonizers of Mars! You arrived on Mars as part of Melon Usk's Mars Three expedition. (Sadly, the Mars One and Mars Two were not successful. The third time's the charm!) Your duty today is to maintain an array of antennas (arranged in a line) that enables the colony to communicate with Earth. Due to fairly strong solar radiation, the antennas tend to malfunction often. Each time it happens you must replace the faulty electronics with spare parts you have in your EdisonY Mars Rover. Unfortunately, the rover has only a limited battery life, so you need to use it sparingly and, in particular, sometimes it is better to simply take the spare part with you and walk to the faulty antenna by yourself instead of driving there.

This gives rise to the following online problem. Let $p_0 = 0$ be the initial position (on the line) of the rover. In each round t:

- You receive a faulty antenna request at location r_t. (The value of r_t is a location on the line.)
- Upon receiving the request r_t, you have to decide which position p_t to move your rover to. (Staying in place, i.e., setting $p_t = p_{t-1}$ is a valid choice). The cost to move is $C \cdot |p_t - p_{t-1}|$, i.e., C times the distance the rover moved, for some $C > 1$.
- Then, you need to service the request and this incurs an additional cost of $2|p_t - r_t|$ corresponding to your needing to walk from the rover to the faulty antenna (and then come back).

(a) [10 points] Consider a strategy in which the rover always stays put. That is, in each round t you choose $p_t = p_{t-1}$. Construct a family of inputs (i.e., sequence of antenna faults) that shows that the competitive ratio of this strategy can be arbitrarily large. (Assume there is an antenna at every integer coordinate of the line.)

(b) [10 points] Consider next a strategy in which you always move the truck all the way to the faulty antenna. That is, $p_t = r_t$, in each round t. Argue that this strategy has a competitive ratio of *at least* $\frac{C}{2}$. Specifically, design a sequence of requests such that the total cost of the strategy is at least $\frac{C}{2}$ larger than the cost of the optimal solution for that sequence.

(c) [10 points] Argue that the competitive ratio of the strategy from point (b) is $O(C)$.

Hint: Use a potential function $\Phi(p_t, OPT_t) = C \cdot |p_t - OPT_t|$ and triangle inequality. Here, OPT_t is the position of the rover in the optimal strategy after it moves in round t.

Problem-5: [10 points] **Connecting Mars**

Melon Usk has set up a colony on Mars that is made up of n stations. The stations are connected by a fiber network that consists of a total of m links, each connecting two stations. Initially, every station can reach every other station through this network.

Melon checks the weather forecast and sees that there is a dust storm approaching! He can see that the dust storm will destroy k of the links in his network (where $k \leq m$). At each time t_i for $1 \leq i \leq k$, where $t_1 < t_2 < \ldots < t_k$, the fiber link between stations u_i and v_i will be destroyed by the dust storm. Once some links are cut, the stations might not all be connected to each other anymore. For each time t_i, Melon wants to calculate the size of the largest subset of stations that can still reach each other through the fiber network at time t_i. (A station is reachable from another station if there exists a path through the network connecting the two stations.)

Design an algorithm to solve this problem in $O(m\alpha(n))$ time. Justify the correctness and runtime of your algorithm.

Problem-6: [20 points] **Nights of the Hash Table**

Melon Usk is organizing a dinner party where n guests are seated around a round table with $N = 2n$ seats, numbered 1 through N clockwise.

The guests arrive one by one, and each guest takes his or her seat before the next guest arrives. To optimize the amount of networking between the guests, Melon has come up with an unusual way of seating them. Namely, to find his or her seat, each guest i first receives a seat number r_i chosen uniformly and independently at random. Then, if seat r_i is not currently occupied, the guest sits there; otherwise, the guest walks clockwise around the table until he or she finds the first unoccupied seat, and sits there.

A *block* is a set of consecutive seats that are all occupied but with the seats before and after it being unoccupied. Let $p_{j,k}$ be the probability that after all the guests have taken their seats there is a block of length k starting at seat j , i.e. that $\{j, \dots, j + k - 1\}$ is a block in the final configuration.

(a) [4 points] Let $E_{j,k}$ be the event that at least k of the random numbers $r_1, \dots, r_n$ given to the guests lie in the set $\{j, \dots, j + k - 1\}$. Argue that $p_{j,k} \le Pr[E_{j,k}]$.

(b) [9 points] Show that $Pr[E_{j,k}] \le \frac{1}{c^k}$ for some constant $c > 1$ independent of the problem parameters.

Hint: Consider random variables X_i, for each guest i, where $X_i = 1$ if $r_i \in \{j, \dots, j + k - 1\}$ and $X_i = 0$, otherwise; and recall that $e > 1$.

(c) [7 points] After all the guests have taken their seats, Melon finally joins them and heads for his favorite seat (which of course is number 1). However, it is quite possible that by that time there is already somebody sitting there because nobody saved Melon's favorite seat!

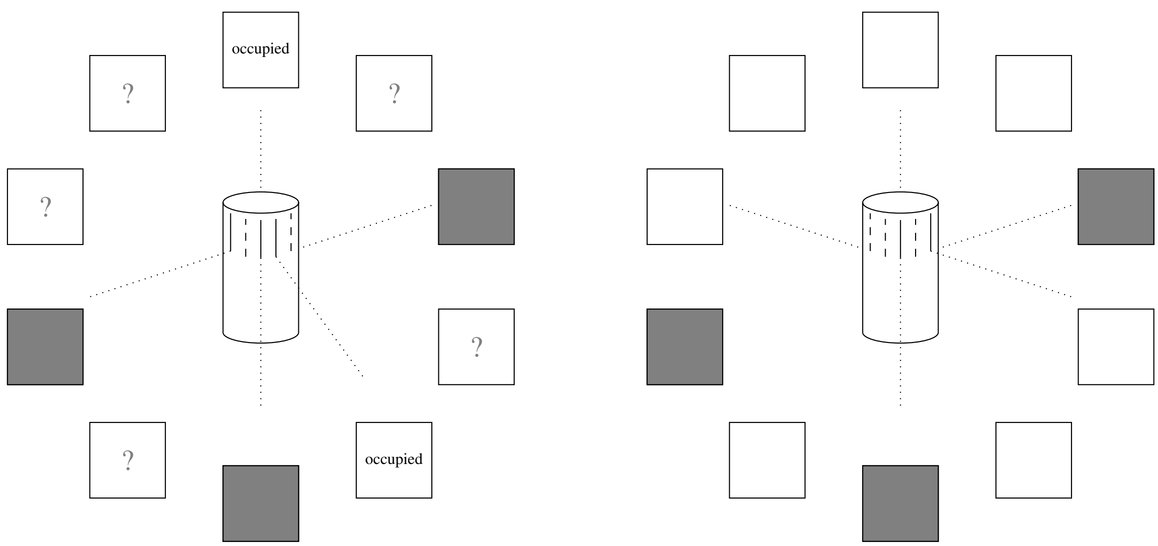
Show that if Melon follows the same protocol for resolving this problem (i.e., walk clockwise until the first unoccupied seat), with probability at least $1 - \frac{1}{N^2}$, the number of the seat he ends up taking will be $O(\log n)$.

Hint: Prove that with probability at least $1 - \frac{1}{N^2}$, there is no block of length $C \cdot \log n$ at the table, for some sufficiently large constant C.

Problem-7: [15 points] **The Room Where It Happened**

As his right hand aide, Detective Baron Hurr takes you to a crime scene where you see a circle of n chairs, some of which have been splattered with paint. As you look to the side, you see a canister of paint with vertical slits along the circumference. Surprisingly, all the slits occur in multiples of $2\pi/n$. You realize that the can was in the middle of the circle when it erupted, causing paint to be shot through the slits. However, some of the chairs were occupied during this time and thus were shielded from paint, but you have no idea which chairs were occupied.

Given the positions of the slits and which chairs are painted, *how many orientations could the canister have been in when it exploded?*



(a) Valid orientation. Chairs with "occupied" must have been occupied. Chairs with "?" could have been occupied.

(b) Invalid orientation. In this orientation, not all painted chairs were shot at.

Figure 1: *Shaded squares are chairs with paint. In valid orientations, all painted chairs are aligned with a slit. (Slits on the back side of the canister are not shown.)*

Detective Hurr marks the canister clockwise with $a_1a_2a_3 \ldots a_n$ where a_i is 1 if the canister has a slit in position i and 0 otherwise. He then marks the chairs clockwise $b_1b_2b_3 \ldots b_n$ where b_i is 1 if the chair has paint on it and 0 otherwise.

In short, paint must have shot towards all painted chairs, but also could have shot towards unmarred chairs. The floor is a complete mess, and you can't tell anything about the trajectories of paint based on the rest of the room. Recall, you want to determine the number of valid orientations of the canister given the state of the room.

Everyone's got their eyes on you. What do you say? Design an algorithm to find the number of valid orientations of the canister in $O(n \log n)$ time.