CS 161 (Stanford, Winter 2023) Section 9

1 Stable Matching

Let's review what we've learned about Stable Matching (Gale-Shapley Deferred Acceptance Algorithm).

(a) Suppose we have doctors A and B, and hospitals X and Y. Construct a set of preferences for these 2 doctors and 2 hospitals in which there is more than one stable matching.

(b) Suppose we have n doctors and n hospitals, and all doctors have different favorite hospitals. With doctors "proposing" first, how many steps does it take for the algorithm to converge?

(c) Suppose that all n doctors have identical preferences over the n hospitals. How many steps does it take for the algorithm to converge (with doctors "proposing" first again)?

(d) We can consider other pairing problems where we do not have "two sides." Consider the problem of pairing 4 students together, where each person has preferences over which of the others they would prefer to have as a project partner. Everyone must pair off. Construct a set of preferences where no stable matching exists.

2 Investing

Suppose you are investing in a single stock a. You want to buy low, then sell high. You have an array A of integers representing future prices of a, and can make one buy followed by one sell. Regardless of the price at which you buy, you will only purchase a single unit. What is the maximum profit you can make on this investment?

1. Design an $O(n \log n)$ divide-and-conquer algorithm to return the maximum potential profit, and justify its runtime.
2. Design an O(n)-time algorithm to solve this problem, and justify its runtime.

3 Quicksand

We are travelling through a marsh which can be mapped to an $M \times N$ grid. The marsh is mostly solid ground, but some parts are quicksand pits located throughout the marsh that are unsafe to travel through. In addition, the locations adjacent (up / down / left / right) to the quicksand pits are also unsafe. At each timestep, you can travel to any of the locations adjacent to your current location in the marsh (diagonal moves are *not* allowed). Design an algorithm that returns a shortest safe path from one side of the marsh to the other (starting at any location in the leftmost column of the grid and ending at any location in the rightmost column), and analyze its runtime.

4 Longest Palindromes

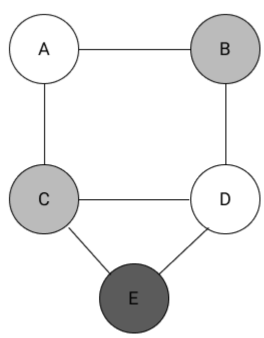
A string is a *palindrome* if it is the same both forwards and backwards. For example, "kayak" is a palindrome, but "canoe" is not. Similarly, "aa" is a palindrome, but "abaa" is not. ("a" is also a palindrome.)

A *subsequence* of a string is any sequence of characters that can be derived from the original string by deleting characters from that string. For example, the subsequences of the string "aid" are "aid", "ai", "ad", 'a", "id", "i", "d", and " ".

Design an algorithm that takes a string and returns the length of the longest subsequence that is a palindrome. Analyze the runtime of your algorithm.

5 Graph Coloring

Minimum graph coloring is an open NP-hard problem for finding the minimum number of colors needed to color all the nodes in the graph such that no nodes of the same color share an edge. Below is an example of a minimum-color graph.



1. Although the problem is NP-hard, we can use greedy algorithms to obtain a pretty good solution. Describe a greedy algorithm that never uses more than d+1 colors, where d is the maximum degree of a vertex in the given graph. Your algorithm should run in $O(V^2 + E)$.
2. Prove by counter example that your greedy algorithm does not always return the correct minimum coloring. Your solution should include a graph, the correct minimum coloring, and the minimum coloring returned by the greedy algorithm.
3. Prove that the graph will return at most a colorization of $d + 1$. (note: You may use proof by induction, but you do not need to for this problem.)

6 Covering a Number Line

Toby the Terrapin walks along a number line, starting at position n. Every second, he moves left with probability p and right with probability $1 - p$. Design an algorithm that computes the probability that Toby is at position x on the number line after n seconds, for all integer values of x (but some specific values for p and n). Discuss the runtime of your algorithm.

7 Minimum Spanning Trees

Given a weighted, undirected, connected graph $G = (V, E)$, with edge weights in the set $W = \{1, 2, 3\}$, design an algorithm to find a minimum spanning tree, and analyze its runtime.

8 Hashing

Assume that H_1 is a valid universal hash family that output within the range 1 through n.

Recall the definition of a universal hash family F: for a randomly chosen h from F, $P(h(x) = h(y) \leq 1/n$, where n is the size of the output space of the functions in F.

Notationally, take $H_x = H_1 + 1$ to mean: "The hash family H_x is composed of each of the hash functions in H_1, but modified so that each function adds one at the end." You can think of each hash family as a single hash function if that's easier.

State whether or not the following new hash families are valid universal hash families.

(a) $H_a(x) = (H_1(x) + 100) \mod n$

(b) $H_b(x) = H_1((x * 2) \mod n)$ (assume that n is divisible by 2).