CSE 102 Spring 2021
Homework Assignment 6

Jaden Liu
University of California at Santa Cruz
Santa Cruz, CA 95064 USA

May 18, 2021

1 HW6

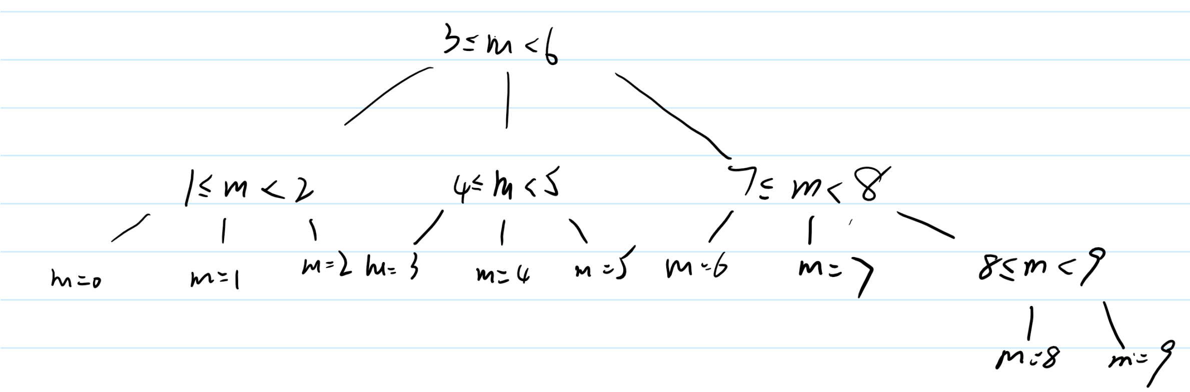
1. Let m be an integer in the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, and consider the following problem: determine m by asking 3-way questions, i.e. questions with at most 3 possible answers. For instance, one could ask which of 3 specific subsets m belongs to.

   a. Give a decision tree argument showing that at least 3 such questions are necessary in worst case. In other words, prove that no correct algorithm can solve this problem by asking only 2 questions in worst case.

   b. Design an algorithm that will solve this problem by asking 3 such questions in worst case. Express your algorithm as a decision tree.

Solution for a. The decision tree will have $\log_3 10$ questions at least, so we need to ask at least 3 questions. Or we can think another way: when we only ask two questions with 3 answers to each question, we have at most 3^2 9 answers. However, we have 10 element in the set, so we cannot get any of the element by asking less than 3 questions. □

Solution for b.



□

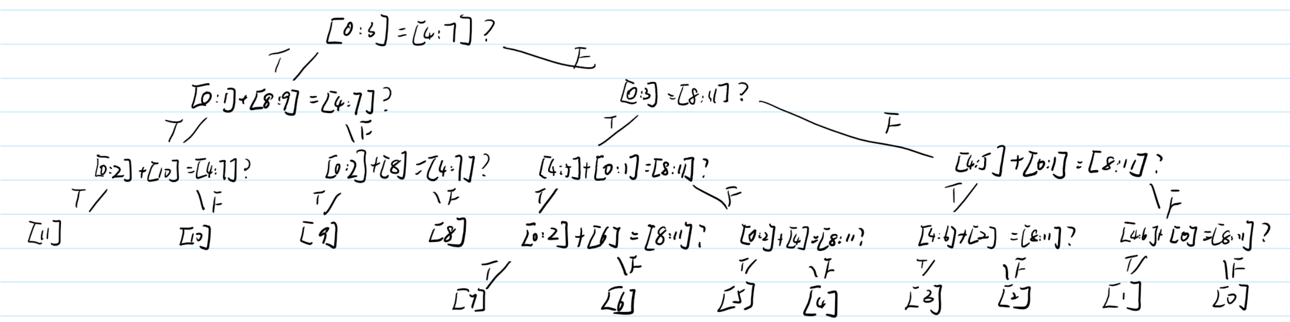
2. **Bar Weighing Problem**
Assume we are given 12 gold bars numbered 1 to 12 where 11 bars are pure gold and one is counterfeit: either gold-plated lead (which is heavier than gold), or gold-plated tin (lighter than gold). The problem is to find the counterfeit bar and what metal it is made of using only a balance scale.

Any number of bars can be placed on each side of the scale, and each use of the scale produces one of three outcomes: either the left side is heavier, or the two sides are the same weight, or the right side is heavier.

a. Give a decision tree lower bound for the (worst case) number of weighings that must be performed by any algorithm solving this problem.
b. Design an algorithm that solves this problem with (worst case) number of weighings equal to the lower bound you found in (a). Present your algorithm by drawing a decision tree, rather than pseudo-code.
c. Alter the problem slightly to allow the possibility that all 12 bars are pure gold. Thus there is one additional possible verdict: "all gold". Make a minor change to your algorithm in part (b) so that it gives a correct answer to this more general problem.

Solution for a. Since $\log_2 12 \approx 3.5850$, we need at least 4 weighings to solve this problem. $\square$

Solution for b.



$\square$

Solution for c. So for our algorithm, we can add another compare process after the first step. When we compare [0:3] to [4:7] and get the result true which implicits that there is a counterfeit in [8:11]. Thus, we need to add one more process in here to determine whether [8:11] has a counterfeit, comparing either [0:3] or [4:7] with [8:11] so that we will increase our minimum comparasions to 4 times. $\square$

3. **Water Jug Problem** (Problem 8-4: page 206 of CLRS 3rd edition)
Suppose that you are given n red and n blue water jugs, all of different shapes and sizes. All red jugs hold different amounts of water, as do the blue ones. Moreover, for every red jug, there is a blue jug that holds the same amount of water, and vice versa.

It is your task to find a grouping of the jugs into pairs of red and blue jugs that hold the same volume of water. To do so, you may perform the following operation: pick a pair of jugs, one red, one blue, fill the red jug with water, and then pour the water into the blue jug. This operation will tell you if the two jugs hold the same amount of water, and if not, which one holds more water. Assume that such a comparison takes one unit of time. Your goal is to find an algorithm that solves this problem. Remember that you may not directly compare two red jugs or two blue jugs.

a. Describe an algorithm that uses $\Theta(n^2)$ comparisons (in worst case) to group the jugs into pairs.
b. Prove a lower bound of $\lceil \log_3(n\ !) \rceil$ for the worst case, and $\log_3(n\ !)$ for the average case number of comparisons to be performed by any algorithm that solves this problem.

Solution for a. First we pick a red jug, then compare with one of the blue jug, stopping until we find a same capacity blue jug. Assume every time finding a pair is the in the worst case, which only make a pair in the last comparison. Thus, it takes $n+(n-1)+(n-2)+\cdots+1=\frac{n(n+1)}{2}=\Theta(n^2)$ comparing times. □

Solution for b. We can change the problem to two array, red jugs are $R[]=[r_1,r_2,\cdots,r_n]$, while the blue jugs are $B[]=[b_1,b_2,\cdots,b_n]$. Then we just sort the two arrays, and the corresponding element in another array is the pair we need. Thus, we need $n!$ arrangement to sort this, and for each comparison, we have three result: equal, less than or greater than. Thus the worst case is $\lceil \log_3 n! \rceil$. Average case needs $\log_3 n!$. □

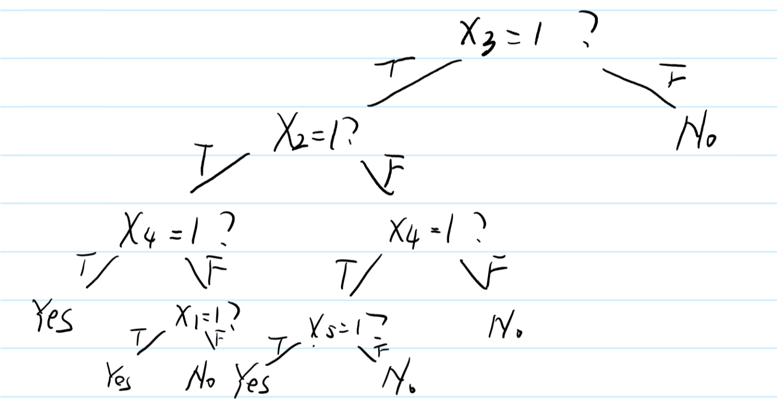
4. Show that at least $\binom{n}{2}$ "adjacency" questions are necessary to determine whether a graph G on n vertices is acyclic. (Hint: use the following adversary strategy. Answer yes to any edge probe, unless that answer would prove the existence of a cycle.)

Solution. We can also think about the adversary strategy like the strategy of minimum spanning trees.[1] Consider the edges one at a time in increasing order of length. If adding an edge would connect the graph, declare it part of the spanning tree; otherwise, throw it away. Keep the process until we reach all $\binom{n}{2}$ edges. It's like a inverse process of Kruskal algorithm. In that case, examines all $\binom{n}{2}$ is necessary. □

5. Let $b = x_1x_2x_3x_4x_5$ be a bit string of length 5, i.e. $x_i \in \{0, 1\}$ for $1 \le i \le 5$. Consider the following problem. Determine whether or not b contains the substring 111. Restrict attention to those algorithms whose only operation is to peek at a bit. Obviously 5 peeks are sufficient. A decision tree argument provides the (useless) fact that at least one peek is necessary.

   a. Design an algorithm for this problem that uses only 4 peeks in worst case. Express your algorithm as a decision tree.
   b. Use an adversary argument to show that 4 peeks are necessary in general.

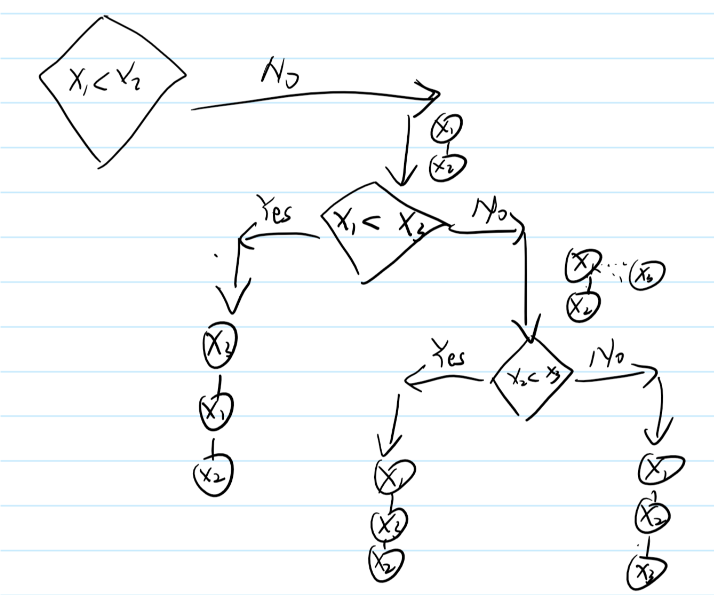
Solution for a. First we need to compare x_3 is 1, if it's not, then there cannot be a substring 111. Then compare either x_2 or x_4 then the other one.



□

Solution for b. Assume we have 00111, then we have yes, no, yes, yes; four answers in total generates there exist a substring 111. □

6. Complete the right side of the "Decision Tree Diagram with Intermediate Results" (for sorting 3 numbers, shared with lecture notes). How many possible outcomes are there? How many leaves are there?



There are three possible outcomes and

hence three leaves.

References

[1] https://courses.engr.illinois.edu/cs473/sp2010/notes/20-adversary.pdf