Homework Assignment # 7

Lower Bounds: Decision Trees and Adversary Argument

Part A

1. **Sorting**: Using decision trees, prove that any comparison based sorting algorithm must take at least $\Omega(n \log n)$ comparisons to sort an array of $n$ integers. What is the minimum number of comparisons needed to sort an array of 5 integers?

2. **Sorting**: Using adversary argument, prove that any comparison based sorting algorithm must take at least $\Omega(n \log n)$ comparisons to sort an array of $n$ integers.

3. **Bit Strings**: Present an adversary argument to prove that any strategy to find out if there are 3 consecutive 1’s in a bit string (consisting of only 0’s and 1’s) of length 5 must peek at 4 bits. The only operation allowed is peeking at a bit.

4. **Acyclic Graph**: Consider an algorithm that tests whether or not a given undirected graph is acyclic by asking questions of the form “Is there an edge between vertex $i$ and vertex $j$?” No other operations are permitted. Provide an adversary argument to prove that the algorithm must enquire about each of the $\binom{n}{2}$ pairs of vertices in an $n$-vertex graph in the worst case.

5. **Bar Weighing**: 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 (heavier than the others) or gold-plated tin (lighter than the others). The problem is to find the counterfeit bar and what metal it is made of using 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, indicating which side is heavier or if they are in balance.)

   (a) Produce a formal decision tree lower bound showing that any algorithm for this problem (using the balance scale to obtain information about the bars) must do at least a certain number of weighings in its worst case.

   (b) Use the insight gained by your lower bound to design an optimal algorithm for the problem using the fewest possible (worst case) number of weighings. (You can use “subroutines” in the description of your algorithm description).
(c) Now consider the same problem where there are 13 bars: 12 solid gold and 1 counterfeit. *Either* come up with an algorithm that correctly solves this 13 bar problem in 3 weighings *or* prove that no algorithm can solve this 13 bar problem using 3 weighings (in the worst case).

**Part B**

1. Prove by adversary argument that any algorithm correctly solving the following problem must examine at least $4\lfloor n/5 \rfloor$ array elements. You may assume that $n$ is a multiple of 5, but your adversary should work for all multiples of 5.

   Given an array $A[1..n]$ of $n$ bits (each element of $A$ is either 0 or 1), answer “yes” if there exists a $j$ such that $1 \leq j \leq n-2$ and $A[j] = A[j+1] = A[j+2] = 1$ (i.e. $A$ contains three consecutive ones), and “no” otherwise.