1 Dumbbell Matching (33 points)

You belong to a gym which has two sets of dumbbells $A$ and $B$, each of which has $n$ dumbbells. You know that there is a correspondence between the sets: for every dumbbell in set $A$ there is exactly one dumbbell in set $B$ that has the same weight, and similarly for every dumbbell in set $B$ there is exactly one dumbbell in set $A$ that has the same weight. You want to perform exercises that require two dumbbells of the same weight. So you want to pair up the dumbbells by weight, i.e., for every dumbbell you want to know which dumbbell from the other set has the exact same weight.

Unfortunately the dumbbells are unsorted and unlabeled, and you can’t tell their weights by looking at them. The only way to compare two dumbbells is to pick them both up simultaneously (one in each hand) and perform a curl. By comparing the strain on your arms, you can tell whether the two dumbbells are the same weight, and if not, which one is heavier. Even more unfortunately, the owner of the gym has a rule that two dumbbells from the same set cannot be used at the same time. So you can compare a dumbbell from set $A$ to a dumbbell from set $B$, but cannot compare two dumbbells from the same set.

Design a randomized algorithm which correctly determines the pairing between the sets and uses only $O(n \log n)$ comparisons in expectation. As always, prove correctness and running time.

2 Costly Median (33 points)

Suppose that you are given $n$ distinct numbers $x_1, x_2, \ldots, x_n \in \mathbb{R}^+$, each of which also has a cost $c_i \in \mathbb{R}^+$ so that $\sum_{i=1}^{n} c_i = 1$. The costly median is defined to be the number $x_k$ such that

$$\sum_{i:x_i<x_k} c_i < \frac{1}{2} \quad \text{and} \quad \sum_{i:x_i>x_k} c_i \leq \frac{1}{2}.$$ 

Give a deterministic algorithm which finds the costly median and has $O(n)$ worst-cast running time (and prove correctness and running time).

3 Algorithms for Sorted Arrays (34 points)

Suppose that you are given a sorted array $A$ of length $n$ and two values $x$ and $y$ with $x < y$. Consider the following problem: determine a) how many elements of $A$ are less than $x$, b) how
many elements of $A$ are at least $x$ and at most $y$, and c) how many elements of $A$ are larger than $y$. Find a function $f(n)$ and prove the following about it.

(a) (17 points) Any algorithm for this problem in the comparison model must have running time $\Omega(f(n))$.

(b) (17 points) There is an algorithm for the problem which runs in time $O(f(n))$ (give such an algorithm and prove running time and correctness).

Note that this is the same function $f(n)$ in each of the subproblems, i.e., you should be finding matching lower and upper bounds.