1. (40 points) Design an $O(m \log n)$ time algorithm that given an unweighted graph $G = (V, E)$ and edge weights $c_e > 0$ for every $e \in E$, outputs the maximum spanning tree: i.e. among all the spanning tree for $G$, the algorithm outputs the one with the maximum total weight of edges (breaking ties arbitrarily).

   Correct algorithm: 25 pts
   Proof of correctness: 15 pts

2. (45 points) Chapter 5, Exercise 1:

   You are interested in analyzing some hard-to-obtain data from two separate databases. Each database contains $n$ numerical values (so there are $2n$ values total) and you may assume that no two values are the same. You’d like to determine the median of this set of $2n$ values, which we will define here to be the $n^{th}$ smallest values.

   However, the only way you can access these values is through queries to the databases. In a single query, you can specify a value $k$ to one of the two databases, and the chosen database will return the $k^{th}$ smallest value that it contains. Since queries are expensive, you would like to compute the median using as few queries as possible.

   Give an algorithm that finds the median value using at most $O(\log n)$ queries.

   Correct algorithm: 25 pts
   Proof of correctness: 10 pts
   Runtime analysis: 10 pts

3. (15 points) Chapter 4, Exercise 6:

   Your friend is working as a camp counselor, and he is in charge of organizing activities for a set of junior-high-school-age campers. One of his plans is the following mini-triathlon exercise: each contestant must swim 20 laps of a pool, then bike 10 miles, then run 3 miles. The plan is to send the contestants out in a staggered fashion, via the following rule: the contestants must use the pool one at a time. In other words, first one contestant swims the 20 laps, gets out, and starts biking. As soon as this first person is out of the pool, a second contestant begins swimming the 20 laps; as soon as he or she is out and starts biking, a third contestant begins swimming and so on.

   Each contestant has a projected swimming time (the expected time it will take him or her to complete the 20 laps), a projected biking time (the expected time it will take him or her to complete the 10 miles of biking), and a projected running time (the expected time it will take him or her to complete 3 miles of running). Your friend wants to decide on a schedule for the triathlon: an order in which to sequence the starts of the contestants. Let’s say that
the completion time of a schedule is the earliest time at which all contestants have completed all three legs of the triathlon, assuming they each spend exactly their projected swimming, biking, and running times on the three parts. (Again, note that participants can bike and run simultaneously, but at most one person can be in the pool at any time.) What’s the best order for sending people out, if one wants the whole competition to be over as early as possible? More precisely, give an efficient algorithm that produces a schedule whose completion time is as small as possible.

Note: As mentioned in the previous homework, in many real life problems, not all parameters are equally important. Also sometimes it might make sense to “combine” two parameters into one. Keep these in mind when tackling this problem.

Hint: The algorithm’s correctness may follow the exchange argument. This question should look familiar.

Correct algorithm: 8 pts
Proof of correctness: 7 pts