

Feel free to work in groups of at most 4 for these - if you have a group of more than 4, please run it by me first. If you do work in a group, please include the names of those that you worked with. **However: each student should submit a separate copy where the solutions have been written by yourself.**

1. (10 points) A particular shipping company facility is looking to optimize their truck loading procedures. Boxes arrive at their facility one by one and **must be loaded into trucks in the order of arrival**. Unfortunately, only one truck can fit in their loading dock at a time. The company currently loads by filling trucks until they reach a box they cannot fit, at which time they send off the current truck and begin loading the next. The company wonders if they might be able to save on the number of trucks used by sending off a truck that is less full, allowing future trucks to potentially be better packed. Convince me that their current strategy is optimal.

**Answer:** Their strategy is optimal. Suppose  $OPT$  is some optimal packing, and look at the first truck  $T$  where it differs from the solution  $S$  produced by this company. Since the company packs trucks to their limit, we know that for this truck  $T$  the company packed  $T$  more than  $OPT$  does. Consider a solution  $OPT'$  that is identical to  $OPT$ , except that it fills up  $T$  (so these extra boxes have been taken from the future trucks they have been packed in in  $OPT$ ). Note that the only truck that has gained more boxes is  $T$  - every other truck has either remained the same or lose some boxes. Since  $S$  does not overfill trucks and in  $OPT'$  we are simply packing  $T$  to the extent that  $S$  does, we are not overfilling  $T$ . Thus we are not increasing the number of trucks we are using, but have created a solution that agrees with  $S$  for one more step. By repeatedly applying this exchange, we will eventually reach the solution  $S$  the company produces, showing that  $S$  is indeed optimal.

2. A railroad path has been constructed, but station locations have not been chosen yet. For our purposes, imagine the railroad as a number line, with some points on the line marked as town locations. Our job is to choose specific points on the line to build train stations; stations need not be collocated with towns, but can be if desired. Every town must be within distance  $R$  of a train station. The goal of the algorithm is to find a minimal collection of locations to build train stations.
  - (a) (5 points) We'll say a train station  $S$  adds a town  $T$  if the town  $T$  is within  $R$  distance of  $S$  and  $T$  is not already within  $R$  distance of a previously chosen train station. Consider the following algorithm: until all towns are added, repeatedly build train stations where you can maximize the number of towns added. Convince me that this algorithm is incorrect.

**Answer:** Consider the following instance, where  $R = 1$  and there are towns at 0, 2, 3, 4, and 5. Then the optimal solution would be to place stations at 1 and 4, as this would cover all the towns. However, the algorithm would first place a station at 3 (to cover 2, 3, and 4), but would then need two more stations to cover 0 and 5. Thus the proposed algorithm is incorrect.

- (b) (10 points) Design a greedy algorithm and convince me that it can be implemented with a polynomial runtime and that it is correct.

**Answer:** Consider the following algorithm  $A$ : sequentially pick the earliest uncovered town, and place a station at distance  $R$  forward from it. Given  $n$  stations, this could be implemented in  $O(n)$  (or  $O(n \log n)$  if not pre-sorted) time, by always keeping track of the location of the last built train station and sequentially going through the town locations to check if they're covered. As for correctness, first observe that  $A$  will certainly cover every town by construction, so we only need to make sure that we don't use more train stations than optimal. Let's suppose  $\text{OPT}$  is some optimal solution and let's look at the first (starting from the left of the number-line) train station that it places differently from  $A$ . Let  $T_1$  denote the location of this next train station in  $\text{OPT}$  and  $T_2$  the next location in  $A$ 's solution.

First, it must be the case that  $T_1 < T_2$ : recall that in  $A$  every train station is placed exactly  $R$  distance past a town; let  $t$  denote the location of this town for this particular train station. Since  $\text{OPT}$  and  $A$  have identical selections before  $T_1$ , if  $T_1$  were past  $T_2$  then  $t$  would be uncovered.

Consider the solution  $\text{OPT}'$  that is identical to  $\text{OPT}$  but instead places the train station at  $T_2$  instead of  $T_1$ . Note that this move doesn't uncover any town - for this to happen, there would have to be a town  $t'$  before  $t$  that is uncovered by the train stations before  $T_1$ . However,  $t'$  would also have to be uncovered in  $A$ 's solution, which as per the observation earlier is impossible. Thus  $\text{OPT}'$  is an optimal solution that coincides with  $A$ 's solution for one more train station. By applying this exchange to each train station, we eventually arrive at  $A$ 's solution, showing that  $A$ 's solution is optimal.

3. For the purposes of this problem, imagine you are a kids' camp counselor in charge of teaching the kids how to play hockey. You have a stock of  $n$  hockey sticks of varying sizes available for the  $n$  kids, also of varying sizes. To make things simpler, let's say that a size 1 stick should be used by a size 1 kid, a size 2 stick with a size 2 kid, and so on. However, any kid can theoretically use any available hockey stick, albeit a bit uncomfortably. You want to figure out a way to distribute the hockey sticks to minimize the total difference between all the kids and their paired sticks.

- (a) (5 points) Consider the algorithm that assigns hockey sticks sequentially with the minimal difference in size possible. So first you assign as many hockey sticks as you can that are perfect matches, then you assign hockey sticks that could be just a difference of 1 in size, then difference 2, and so on until all the hockey sticks have been assigned. Convince me that this algorithm is not correct.

**Answer:** Suppose that we have a size 1 stick, a size 4 stick, a size 3 kid, and a size 6 kid. Then the algorithm will first pair off the size 4 stick to the size 3 kid (for a difference of 1), forcing the size 6 kid to take the size 1 stick, giving a total difference of 6. Optimally, we could assign the 1 stick to the size 3 kid and size 4

stick to the size 6 kid, giving us a total difference of 4.

- (b) (10 points) Design a greedy algorithm and convince me that it can be implemented with a polynomial runtime and that it is correct.

**Answer:** We will consider the algorithm  $A$  that sequentially assigns the smallest hockey stick to the smallest kid, the second smallest to the second smallest, and so on. For an instance with  $n$  kids and  $n$  hockey sticks, this can be implemented in  $O(n)$  (or  $O(n \log n)$  if not pre-sorted) time by simply iterating through the two sorted lists simultaneously. As for correctness, let  $\text{OPT}$  be some optimal assignment and look at smallest hockey stick and kid that are not assigned to each other in  $\text{OPT}$ . Specifically, let  $k$  be this kid and  $h$  this hockey stick, let  $h'$  be the hockey stick  $\text{OPT}$  assigns to  $k$  and let  $k'$  be the kid that is assigned  $h$ . Let  $\text{OPT}'$  denote the solution that is identical to  $\text{OPT}$  except the  $k$  and  $k'$  trade sticks. Following are all the possible orderings (not necessarily drawn to scale) of  $h, k, h',$  and  $k'$ . The red lines denote the difference in  $\text{OPT}$ , while the greens denote the differences in  $\text{OPT}'$ . Since the green lines have total difference at most the red in each scenario, this shows that  $\text{OPT}'$  is an optimal solution. Since it agrees with  $A$  for an additional step, by repeatedly applying this exchange we will eventually reach the solution produced by  $A$ , showing that  $A$  is optimal.

