Due: Tuesday, February 7, 2023, 11:59pm

Submit online at (clickable link): `https://forms.gle/oM9ErRraCGxvNMma7`

1 Greedy Trick or Treat

Consider the following Trick-or-Treat problem. You live on a street with n houses. You have a bag to carry your Halloween candy, but your bag can carry at most K ounces of candy. You are given for each house i, an integer weight w_i (in ounces) of the candy that the people in house i are handing out. Each house gives out one piece of candy. You can visit each house at most once. Your problem is to collect the largest number of pieces of candy from the n houses without exceeding the capacity of your bag.

Now consider the following greedy algorithm. Sort the houses according to the weight of the candy they are providing. Collect candy from the houses starting from the house that provides the *lightest* candy, then the next lightest, ... until your bag is full.

Notes:

- The greedy algorithm for this problem has already been specified. You must prove that the given algorithm produces the optimum solution. Do not consider any other algorithms.
- Demonstrate that you are able to use a swapping lemma argument to prove the optimality of the greedy algorithm. Do not turn this into an algebra or calculus problem.

Questions:

1. Describe the set of feasible solutions for a particular instance of this problem. What makes a solution feasible? What does the objective function measure?
2. State a swapping lemma that can be used to prove that this greedy strategy produces the optimum solution.
3. Argue that applying a single swap (as described in your swapping lemma) to a non-greedy feasible solution produces another feasible solution.
4. Argue that applying a single swap (as described in your swapping lemma) to a non-greedy feasible solution does not decrease its objective value.
5. Prove that the greedy algorithm produces a feasible solution with the optimum objective value.

   Note: You must prove that it is not possible to collect more candy than the greedy algorithm. Do not appeal to any unproven general principles. You must use in your proof the swapping lemma you stated in Question #2 above.

2 Elves at Helm's Deep

It is after the Battle of the Hornburg. The main characters have ridden off to Isengard, and you have been ordered to make sure Helm's Deep is secure. Since the Deeping Wall has been blown to bits, you decide to post guards further out at Helm's Dike instead. Helm's Dike is a long earthen wall that runs across full length of the valley of Deeping-coomb. The dike is quite steep and as tall as twenty feet in some places. However, you have found n places along the dike that may be vulnerable to attack. These are located $x_1, x_2, \ldots, x_n$ yards from the western end of the dike. Your plan is to deploy elven archers along the dike. (Tolkien elves have keen eyesight, are excellent archers, don't sleep and are sort of immortal.) You figure that if you can place an elf within 50 yards of every vulnerable location then the dike will be well fortified.

Now, these elves are not actually under your command. They just came by to help out. Their leader, Haldir, met an unfortunate end during the battle, so they might accede to your request. In any case, you want to ask as few of them as possible to stand guard. (The whole idea of asking semi-immortal beings to stand guard is somewhat awkward, but they don't need sleep!) Fortunately, you have taken algorithms and can find the minimum number of elven archers needed to accomplish this task.

Questions:

1. Describe the set of feasible solutions for a particular instance of this problem. What makes a solution feasible? What does the objective function measure?

2. Devise then describe a greedy algorithm that will find the locations to place the minimum number of elven archers along Helm's Dike so that each of the n vulnerable locations are within 50 yards of at least one elven archer.

3. State a swapping lemma that can be used to prove that this greedy strategy produces the optimum solution.

4. Argue that applying a single swap (as described in your swapping lemma) to a non-greedy feasible solution produces another feasible solution.

5. Argue that applying a single swap (as described in your swapping lemma) to a non-greedy feasible solution does not increase its objective value.

6. Prove that the greedy algorithm produces a feasible solution with the optimum objective value.

   Note: You must show that it is not possible to cover all points of vulnerability using fewer elves than the greedy algorithm uses. Do not appeal to any unproven general principles. You must use in your proof the swapping lemma you stated in Question #3 above.