APS Homework 2: Greedy Method
Optional Challenge Problems

Problem 1: Clock Divisibility

Your friend Bob is a somewhat weird guy. He has two obsessions in his life: his digital clock and divisibility. His digital clock displays the time in the 24-hour format (i.e., the first minute of a day is 00:00, and the last minute of a day is 23:59), and one of Bob's favorite hobbies is to pick a random number $1 \leq x \leq 9$ and just watch his clock until all 4 digits of the display are divisible by x (assume 0 is divisible by every number).

Given an arbitrary number x (between 1 and 9) and an arbitrary current time y in the 24-hour format described above, how many minutes will Bob have to wait until the clock displays a time such that all 4 digits of the time are perfectly divisible by x? For example, if x is 3 and y is 03:23, the answer is 7 minutes, because 7 minutes after y would be 03:30, and all 4 digits of this resulting time are divisible by x.

Problem 1a: Given an integer x (between 1 and 9) and a current time y (in the 24-hour format), describe a greedy algorithm for computing how long Bob has to wait until the clock displays a time such that all 4 digits are divisible by x.

Problem 1b: Prove that the algorithm you provided in *Problem 1a* is correct for any integer $1 \leq x \leq 9$ and any time y.

Problem 2: Wizard's Chess

A chess board is an 8×8 grid. One specific chess piece, the knight, moves in "L" shapes: it moves 2 spaces in one direction and 1 space in a perpendicular direction (where directions are North, South, East, and West). In the completely true documentary *Harry Potter and the Sorcerer's Stone*, the protagonists Harry Potter, Ron Weasley, and Hermione Granger must work together to win a game of life-size Wizard's Chess. Ron chooses to be the knight. Towards the end of the game, Ron sees where he must go in order for his friends to win the game.

Let (x_0, y_0) denote Ron's current position, and let (x_1, y_1) denote the final position where Ron must go, where (0,0) is the bottom-left corner of the board, (7,0) is the bottom-right, (0,7) is the top-left, and (7,7) is the top-right. What path can Ron take to go from (x_0, y_0) to (x_1, y_1) in the least number of steps?

Problem 2a: Describe a greedy algorithm to move Ron from (x_0, y_0) to (x_1, y_1) in the least number of steps.

Problem 2b: Prove that the algorithm you provided in *Problem 2a* is correct for any arbitrary points (x_0, y_0) and (x_1, y_1).

Problem 3: Rearranging the Letters of a String

Remember your friend Bob from before? It turns out that he's even weirder than you thought. For some reason, in addition to watching his digital clock all day, Bob loves to rearrange the letters of strings to make them lexicographically smaller. For example, the string *aaaba* is lexicographically (i.e., alphabetically) smaller than the string *abaaa*.

Given a string x and an integer k, Bob is allowed to rearrange up to k letters in x, and he wants an algorithm that will give him the lexicographically smallest string possible. For example, if x is *ebadc* and k is 3, Bob can do the following:

1. Cut out indices 0 (*e*), 2 (*a*), and 4 (*c*), resulting in _*b*_*d*_
2. Put the *a* in index 0, the *c* in index 2, and the *e* in index 4, resulting in *abcde*

Problem 3a: Describe a greedy algorithm where, given a string x and an integer k, you can cut out k letters from x and put them back into the the now-open slots to yield the lexicographically smallest possible resulting string.

Problem 3b: Prove that the algorithm you provided in *Problem 3a* is correct for any arbitrary string x and any arbitrary integer $1 < k \leq |x|$.