Greedy Algorithms

Assignment Questions

PW SKILLS

Q1. Find the minimum sum of Products of two arrays of the same size, given that k modifications are allowed on the first array. In each modification, one array element of the first array can either be increased or decreased by 2.

Examples:
Input : a[] = {1, 2, -3}
b[] = {-2, 3, -5}
k = 5
Output : -31
Explanation:
Here n = 3 and k = 5.
So, we modified a[2], which is -3 and increased it by 10 (as 5 modifications are allowed).
Final sum will be :
(1 * -2) + (2 * 3) + (7 * -5)
-2 + 6 - 35
-31
(which is the minimum sum of the array with given conditions)

Input : a[] = {2, 3, 4, 5, 4}
b[] = {3, 4, 2, 3, 2}
k = 3
Output : 25
Explanation:
Here, total numbers are 5 and total modifications allowed are 3. So, modify a[1], which is 3 and decreased it by 6 (as 3 modifications are allowed).
Final sum will be :
(2 * 3) + (-3 * 4) + (4 * 2) + (5 * 3) + (4 * 2)
6 – 12 + 8 + 15 + 8
25
(which is the minimum sum of the array with given conditions)

(Easy)

Q2. You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.

Examples:
Input: start[] = {10, 12, 20}, finish[] = {20, 25, 30}
Output: 0 2
Explanation: A person can perform at most two activities. The maximum set of activities that can be executed is {0, 2} [These are indexes in start[] and finish[]]

Input: start[] = {1, 3, 0, 5, 8, 5}, finish[] = {2, 4, 6, 7, 9, 9};
Output: 0 1 3 4

Explanation: A person can perform at most four activities. The maximum set of activities that can be executed is {0, 1, 3, 4} [These are indexes in start[] and finish[]]

Q3. There are n gas stations along a circular route, where the amount of gas at the ith station is gas[i].You have a car with an unlimited gas tank and it costs "cost[i]" of gas to travel from the ith station to its next (i + 1)th station. You begin the journey with an empty tank at one of the gas stations.Given two integer arrays gas and cost, return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1. If there exists a solution, it is guaranteed to be unique.

Example 1:
Input: gas = [1,2,3,4,5], cost = [3,4,5,1,2]
Output: 3
Explanation:
Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank = 0 + 4 = 4
Travel to station 4. Your tank = 4 - 1 + 5 = 8
Travel to station 0. Your tank = 8 - 2 + 1 = 7
Travel to station 1. Your tank = 7 - 3 + 2 = 6
Travel to station 2. Your tank = 6 - 4 + 3 = 5
Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3.
Therefore, return 3 as the starting index.

Example 2:
Input: gas = [2,3,4], cost = [3,4,3]
Output: -1
Explanation:
You can't start at station 0 or 1, as there is not enough gas to travel to the next station.
Let's start at station 2 and fill up with 4 unit of gas. Your tank = 0 + 4 = 4
Travel to station 0. Your tank = 4 - 3 + 2 = 3
Travel to station 1. Your tank = 3 - 3 + 3 = 3
You cannot travel back to station 2, as it requires 4 units of gas but you only have 3.
Therefore, you can't travel around the circuit once no matter where you start.

Q4. You have a long flowerbed in which some of the plots are planted, and some are not. However, flowers cannot be planted in adjacent plots.Given an integer array flowerbed containing 0's and 1's, where 0 means empty and 1 means not empty, and an integer n, return if n new flowers can be planted in the flowerbed without violating the no-adjacent-flowers rule.

Example 1:
Input: flowerbed = [1,0,0,0,1], n = 1
Output: true

Example 2:
Input: flowerbed = [1,0,0,0,1], n = 2
Output: false

Q5. Given an array of intervals where intervals[i] = [starti, endi], return the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping.

Example 1:
Input: intervals = [[1,2],[2,3],[3,4],[1,3]]
Output: 1
Explanation: [1,3] can be removed and the rest of the intervals are non-overlapping.

Example 2:
Input: intervals = [[1,2],[1,2],[1,2]]
Output: 2
Explanation: You need to remove two [1,2] to make the rest of the intervals non-overlapping.

Example 3:
Input: intervals = [[1,2],[2,3]]
Output: 0
Explanation: You don't need to remove any of the intervals since they're already non-overlapping.