Algorithms Laboratory (CS29203)
Assignment 3: Greedy algorithm
Department of CSE, IIT Kharagpur

1st September 2022

Question-1

Count Dracula is the leader of the vampire community, and he seriously wants to impose some rules to prey upon innocent victims by sucking their blood and make them also a dracula. In this way, the dracula cummunity will increase its size, and will eventually capture the whole world. However due to the power restriction, one dracula can't turn more than one innocent person to a dracula within some specified timeline. That is why *Count Dracula* wants to estimate how much will be population of the dracula community by imposing the rules. The rule is that, one dracula can suck blood of only one person who is within some predefined distance d from him. *Count Dracula* is secretly sitting in the Algorithms-I course class at IITKgp and have learned the Greedy algorithms, and he will use this strategy to do the estimation.

In the vision of a dracula, they can only see linear arrangement of people and dracula (like an array). For example if we denote an innocent person as '0' and a dracula as '1', then the following is a valid arrangement: $(0, 1, 0, 1, 0)$. According to the arrangement, the sequence is (person, dracula, person, dracula, person). Let the distance d is 3. Then the dracula at index 1 can catch people in the range $[i - d, i + d] = [1 - 3, 1 + 3] = [-2, 4]$. So he can catch the person who is at index 2. Next, the dracula at index 3 can catch people in the range $[i - d, i + d] = [3 - 3, 3 + 3] = [0, 6]$. So he can catch the person who is at index 0. However, the (lucky) person who is at index 4 will not be caught because the draculas at indices 1 and 3 are already catching one person. Hence, the total number of catch is 2. Note that the optimal solution need not be necessarily unique (there can be different combination(s) of people that will give the same total number of captures).

Your task is to help *Count Dracula* to compute the number of captures by writing a computer program for him (otherwise he will capture you too!). You have to implement a greedy algorithm to solve this problem where your program will take the sequence of person and dracula as a binary array (0=innocent person, 1=dracula), the specified distance as an integer, and return the number of catches as an integer.

Example 1:

```
Enter the binary sequence of people and dracula: 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1
Enter the distance d: 1
---------------
Total number of captures = 4
```

Example 2:

```
Enter the binary sequence of people and dracula: 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1
Enter the distance d: 2
---------------
Total number of captures = 5
```

Example 3:

```
Enter the binary sequence of people and dracula: 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1
Enter the distance d: 4
---------------
Total number of captures = 6
```

Question-2

The sweet shop in the Tech market has got an order for making various types sweets for a meeting event in the CSE department. They have to make n different types of varieites of sweets within some specified timeline. The order is placed in such a way that they need to prepare each variety of sweet within some speficied deadline (that is, if that variety of sweet is delivered, then it has to be delivered on or before that time). There are different costs associated in making the different varieties, and the shop owner wants to minimize the total cost in order to have the maximum profit. Assume that each of the variety of sweet item takes same amount of time to prepare (which is 1 hour). Then the goal is to prepare maximum types of sweet varieties within the deadline, and minimize the total cost of preparation.

For example, consider the following scenario of the order:

Sweet variety #	Latest time to deliver	Cost (in Rupees)
1	9 PM	150
2	2 PM	20
3	5 PM	180
4	7 PM	10
5	4 PM	250
6	2 PM	200
7	5 PM	80
8	7 PM	100
9	4 PM	120
10	3 PM	50

In order minimize the total cost, the optimal choice of sweet varieties are: $3, 2, 10, 9, 7, 8, 4, 1$, and the total cost is Rs. 710.

Your task is to implement a greedy algorithm to solve this problem. The input to your program should be two (integer) arrays: one array listing the latest time of the items (sequentially), and the other array listing the cost of preparing each item. The output of the program will be list of optimally selected sweet ietms, and the total cost of preparing (which will be the minimum).

Example:

```
Enter the number of sweet varieties: 10
Enter the latest time to deliver each item: 9, 2, 5, 7, 4, 2, 5, 7, 4, 3
Enter the cost (in Rupees) of preparing each item: 150, 20, 180, 10, 250, 200, 80, 100, 120, 150
---------------
The optimal choice of sweet ietms are: 3, 2, 10, 9, 7, 8, 4, 1
The total cost of preparing the items: Rs. 710
```