Home Assignment 2
DSA-Refresher 2021

- **You are not supposed to discuss solutions with your classmates or refer to any online resources. Strict Plagiarism policy applies.**
- **Naming Convention: Submit a pdf containing answers to given problem, Naming Convention for file is Name_A2_DSARef21.pdf**
- **Please start early to complete within time.**

1. Placement Sort:

IIITD 2019 batch has been placed. The placement committee wants to upload the recent placement records in IIITD official website. But there is a problem, the list of placed students has random entries.So it has become very difficult to do some quick analysis. You have to perform 2 tasks to help them.

Task1: Sort the records in increasing order by Package. If two students got the same package then their order should be the same as the initial list.

Task2: Perform some functionality that takes a list which we get from task 1 and package(let say x) and returns the number of students who got package ‘x’.

Write a java program to perform these two tasks. Try to achieve minimum time complexity for both the tasks.

State the algorithm used along with time and space complexity for both task1 and task2.

Input Format: First line contains ‘n’ total number of placed students. Next n line contains Rollnumber , Name and package of the student separated by space. Next line contains ‘k’ the number of test cases for the 2nd task. Next ‘k’ lines contain packages.

Output Format: first ‘n’ line contains the record in sorted manner. Next ‘k’ lines show the number of students who got package ‘x’

Note: 10^{5} `<= Package <=` 10^{7}

Example1:

Input:

```
6
MT19112   Shivansh    3000000
MT19100   Akanksha    2000000
MT1912    Divya       3000000
MT1923    Shailja     2000000
MT1911    Anupam      3000000
MT1915    Ritika      2500000
3
```

```
2000000
3000000
1550000
```

Output:

```
MT19100    Akanksha        2000000
MT1923     Shailja         2000000
MT1915     Ritika          2500000
MT19112    Shivansh        3000000
MT1912     Divya           3000000
MT1911     Anupam          3000000
2
3
0
```

2. Deadline Sort:

Ram is a student at IIIT Delhi and has to do a lot of work. He has finished *k* homeworks and the deadlines of these homeworks lie between *0* to $k^2 - 1$.

Write a Java program to help Ram sort these homeworks in *non-decreasing* order of deadlines with **expected time and space complexity O(k).**

State the algorithm used to solve this problem.

Input Format: First line contains integer **k** denoting the number of homeworks. Second line contains **k** space separated numbers which denote the deadlines.

Output Format: Output **k** space separated sorted deadlines.

Example -

Input

```
5
13 24 1 17 8
```

Output

```
1 8 13 17 24
```

3. Helping Hands:

Alex runs a charitable organization which tries to help as many people as possible. His organization takes help from some local vendors to distribute food and other amenities to the people. Let us assume he wants to feed n people (ranging from 0 to n-1) and he has an array helpers[] of size n. Each value in the array denotes the range of that vendor. For each index k, the

vendor can feed people from (k - helpers[k]) to (k + helpers[k]) inclusive on both sides. If at any index the range is -1 means that the vendor is out of stock and cannot feed anyone.
Your goal is to write a Java Program to help Alex find the **minimum** number of vendors which will be able to feed all **n** people.
State the time and space complexity of your approach.

Input Format: First line contains integer **n** denoting the number of people to be helped and the number of vendors. Second line contains **n** space separated numbers which form the array **helpers[]**.

Output Format: Output contains a single integer equal to the minimum number of vendors required. In case all people cannot be fed return -1.

Expected Time Complexity: O(n * log(n)).

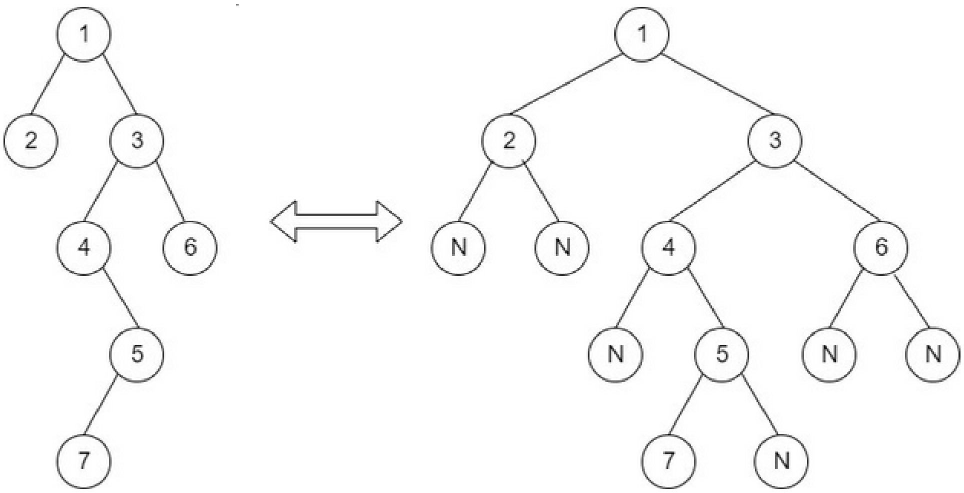
Example -
Input
6
-1 2 2 -1 0 0
Output
2
Explanation:
Here vendors at index 2 and 5 can help to feed everyone. Range of vendor 2 is [0, 4] and range of vendor 5 is [5, 5].

Input
3
0 -1 0
Output
-1
Explanation:
Here no vendor can feed the person at index 1. Range of vendor 0 is [0, 0] and range of vendor 2 is [2, 2].

4. Free Ryloth:

In a galaxy far far away, Ryloth is under the occupation of The Galactic Empire. However, the rise of Luke Skywalker is a ray of hope to the entire galaxy. The rebels have contacted their trusted base (say *informant*) about the plans to free Ryloth. However, to avoid detection each house must spread the information only to houses connected directly to them in one hour. Assume that the houses in Ryloth form a binary tree, where each house represents a node. So *each node can only pass the information to parent and child nodes*. Han Solo and the rebels must wait until each house is aware of the attack so that all the people can help them to free Ryloth. Your goal is to write a Java Program to find the number of hours it will take to spread the information to all the houses of Ryloth.

Input Format: First line contains a string representing the binary tree structure of houses in Ryloth. Second line contains a number **informant** who got the information from the rebels.



For the above tree, the string will be: 1 2 3 N N 4 6 N 5 N N 7 N

Format of Tree Input

Output Format: Output contains a single integer equal to the number of hours the rebels must wait before their attack.

Expected Time Complexity: O(number of nodes in tree).

Example -

Input

1 2 3 N 4 5 6 N N N 7
5

Output

4

Explanation:

Here node 5 takes 1 hour to tell node 3 and node 7. Node 3 tells node 1 and 6 in the second hour. In the third hour, node 1 tells node 2. Finally node 2 tells node 4 in the fourth hour. So rebels must wait 4 hours before attacking.

Input

1 2 3 N N 4 6 N 5 N N 7 N
3

Output

```
3
```