Hackerland Radio Transmitters ☆

143/563 challenges solved
Rank: 4249 | Points: 3369.07

You have successfully solved Hackerland Radio Transmitters Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

Hackerland is a one-dimensional city with houses aligned at integral locations along a road. The Mayor wants to install radio transmitters on the roofs of the city's houses. Each transmitter has a fixed range meaning it can transmit a signal to all houses within that number of units distance away.

Given a map of Hackerland and the transmission range, determine the minimum number of transmitters so that every house is within range of at least one transmitter. Each transmitter *must* be installed on top of an existing house.

For example, assume houses are located at $x = [1, 2, 3, 5, 9]$ and the transmission range $k = 1$. 3 antennae at houses 2 and 5 and 9 would provide complete coverage. There is no house at location 7 to cover both 5 and 9. Ranges of coverage, are $[1, 2, 3]$, $[5]$, and $[9]$.

Function Description

Complete the *hackerlandRadioTransmitters* function in the editor below. It must return an integer that denotes the minimum number of transmitters to install.

hackerlandRadioTransmitters has the following parameter(s):

- *x*: integer array that denotes the locations of houses
- *k*: an integer that denotes the effective range of a transmitter

Input Format

The first line contains two space-separated integers n and k, the number of houses in Hackerland and the range of each transmitter.
The second line contains n space-separated integers describing the respective locations of each house $x[i]$.

Constraints

- $1 \le n, k \le 10^5$
- $1 \le x[i] \le 10^5$
- There may be more than one house at the same location.

Subtasks

- $1 \le n \le 1000$ for 50% of the maximum score.

Output Format

Print a single integer denoting the minimum number of transmitters needed to cover all of the houses.

Sample Input 0

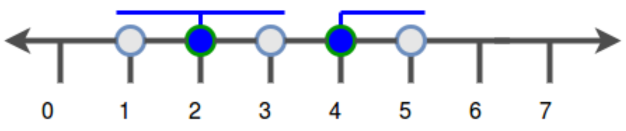
```
5 1
1 2 3 4 5
```

Sample Output 0

```
2
```

Explanation 0

The diagram below depicts our map of Hackerland:



We can cover the entire city by installing 2 transmitters on houses at locations 2 and 4.

Sample Input 1

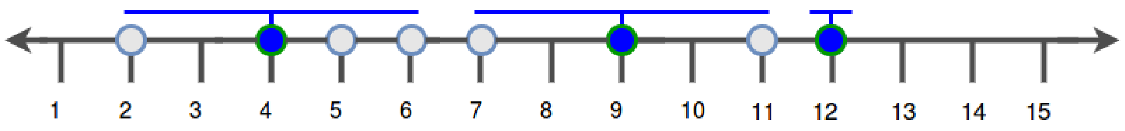
```
8 2
7 2 4 6 5 9 12 11
```

Sample Output 1

```
3
```

Explanation 1

The diagram below depicts our map of Hackerland:



We can cover the entire city by installing 3 transmitters on houses at locations 4, 9, and 12.

Current Buffer (saved locally, editable) Java 7

```java

import java.util.Arrays;
import java.util.Scanner;

public class TwoStreamlined {

    static int hackerlandRadioTransmitters(int[] locationsOfHouses, int rangeOfTransmitter) {
        int minNumberOfTransmitters = 0;
        int currentDistance = 0;
        int index = 0;
        Arrays.sort(locationsOfHouses);

        while (++index < locationsOfHouses.length) {

            currentDistance = locationsOfHouses[index] - locationsOfHouses[index - 1];
            while (currentDistance <= rangeOfTransmitter) {
                if (index < locationsOfHouses.length - 1) {
                    index++;
                    currentDistance += locationsOfHouses[index] - locationsOfHouses[index - 1];
                } else {
                    break;
                }
            }

            currentDistance = locationsOfHouses[index] - locationsOfHouses[index - 1];
            while (currentDistance <= rangeOfTransmitter) {
                if (index < locationsOfHouses.length - 1) {
                    index++;
                    currentDistance += locationsOfHouses[index] - locationsOfHouses[index - 1];
                } else {
                    break;
                }
            }

            boolean lastHouses_NotCoveredByTheLastTransmitter = (index == locationsOfHouses.length - 1
                    && currentDistance > rangeOfTransmitter);

            if (lastHouses_NotCoveredByTheLastTransmitter) {
                minNumberOfTransmitters += 2;
            } else {
                minNumberOfTransmitters++;
            }
        }

        if (minNumberOfTransmitters == 0) {
            minNumberOfTransmitters = 1;
        }

        return minNumberOfTransmitters;

    }

    public static void main(String[] args) {

        Scanner reader = new Scanner(System.in);
        int numberOfHouses = reader.nextInt();
        int rangeOfTransmitter = reader.nextInt();
        int[] locationsOfHouses = new int[numberOfHouses];

        for (int i = 0; i < numberOfHouses; i++) {
            locationsOfHouses[i] = reader.nextInt();
        }
        reader.close();

        int result = hackerlandRadioTransmitters(locationsOfHouses, rangeOfTransmitter);
        System.out.println(result);
    }
}
```

Line: 68 Col: 2

Upload Code as File ☐ Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

✓ Testcase 0 ✓ Testcase 1 ✓ Testcase 2 ✓ Testcase 3 ✓ Testcase 4 ✓ Testcase 5

✓ Testcase 6 ✓ Testcase 7 ✓ Testcase 8 ✓ Testcase 9 ✓ Testcase 10 ✓ Testcase 11

✓ Testcase 12 ✓ Testcase 13 ✓ Testcase 14 ✓ Testcase 15 ✓ Testcase 16

✓ Testcase 17 ✓ Testcase 18 ✓ Testcase 19 ✓ Testcase 20 ✓ Testcase 21

✓ Testcase 22 ✓ Testcase 23 ✓ Testcase 24 ✓ Testcase 25 ✓ Testcase 26

✓ Testcase 27 ✓ Testcase 28 ✓ Testcase 29 ✓ Testcase 30

Input (stdin) Download

```
5 1
1 2 3 4 5
```

Expected Output Download

```
2
```

Compiler Message

```
Success
```

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature