Monster World

locked

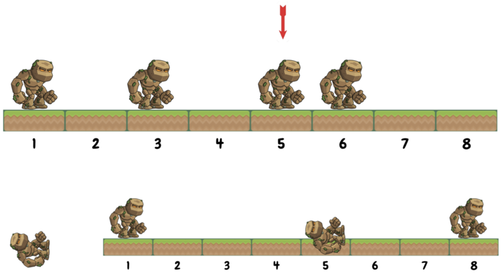
Problem | Submissions | Leaderboard | Discussions

Submissions: 128
Max Score: 100
Difficulty: Medium

Rate This Challenge:

More

Иван играе на любимата си компютърна игра. За съжаление има затруднения с преминаването на едно от последните нива и се нуждае от вашата помощ. Нивото се състои от множество чудовища намиращи се на голяма платформа. Иван трябва да премахне всички чудовища, като използва своя бластер, който му позволява да унищожи всички чудовища на дадена позиция. Бластера е толкова мощен, че избутва всички останали чудовища с x единици разстояние от мястото на взрива. Тъй като началото на платформата е голяма пропаст, то Иван може да се отърве от чудовищата и като ги избута в пропастта с помощта на бластера. Вашата задача е да помогнете на Иван да премине нивото с възможно най-малък брой използвания на бластера.



На горния пример бластера е използван на позиция 5, а стойността на x е 2.

Input Format

- n - брой чудовища
- x - разстоянието, на което бластера избутва чудовищата от центъра на взрива.
- s_i - позицията, на която се намира i-тото чудовище

На първия ред ще получите числата n и x. На втория ред ще получите числата s_i. Всички числа ще са разделени с един интервал.

Constraints

- $1 \leq n \leq 2 * 10^5$
- $1 \leq x \leq 2 * 10^5$
- $1 \leq s_i \leq 10^6$

Output Format

Изведете числото y - минималният брой използвания на бластера, нужни за преминаването на нивото.

Sample Input 0

```
3 2
1 3 5
```

Sample Output 0

```
2
```

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

```
4 2
5 3 1 6
```

Sample Output 2

```
2
```

Current Buffer (saved locally, editable) C++

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
#include<random>
using namespace std;
void deleteEqualMembers(int* arr, int& creatureNumber) {
int* temp=new int[creatureNumber];
    int j=0;
    for(int i=0;i<creatureNumber;i++){
        if (arr[i] != arr[i+1])
            temp[j++] = arr[i];
    }

    for (int i=0; i<j; i++){
        arr[i] = temp[i];
        }
    creatureNumber=j;

}
void countingSort(int* arr, int size) {
    int maxNumber = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] > maxNumber) {
            maxNumber = arr[i];
        }
    }

    int * helpArr = new int[maxNumber + 1];
    for (int i = 0; i < size; i++) {
        helpArr[arr[i]]++;
    }
    int sortedIndex = 0;
    for (int i = 0; i < size; i++) {
        while (helpArr[sortedIndex] == 0) {
            sortedIndex++;
        }

        arr[i] = sortedIndex;
        helpArr[sortedIndex]--;
    }
    delete[] helpArr;
}

int minBlasterShots(int* creatures, int& blasterPower, int creatureNumber) {
    int counter = 0;
    int lastElement=creatureNumber-1;;
    while (creatureNumber > 0) {
        ++counter;
        creatures[lastElement--]=-1;
        --creatureNumber;
        creatures[lastElement]-=blasterPower*counter;
        if(creatures[lastElement]<=0){
            break;
        }
        }



    return counter;
}

int main() {
    int creatureNumber;
    cin >> creatureNumber;
    int blasterPower;
    cin >> blasterPower;
    vector<int> creatures;
    int creaturePos;
int* creaturesArr=new int[creatureNumber];
    for (int i = 0; i < creatureNumber; i++) {
        cin >> creaturesArr[i];
    }
    countingSort(creaturesArr, creatureNumber);
    deleteEqualMembers(creaturesArr, creatureNumber);
    cout << minBlasterShots(creaturesArr, blasterPower, creatureNumber);
    delete[] creaturesArr;
    return 0;
}


```

Line: 1 Col: 1

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