Block Adventure 题解

> 计算学部十大打卡活动——“龙舞编程新春会”编程打卡（2月1日）
>
> 题目为Block Adventure

题面翻译

题目描述

你在玩一个游戏，已知在你面前有n列砖块，你的背包中有m个砖块，第i列有h_i个砖块。

在第i列你可以进行下列操作

- 如果你的背包中有砖块，你可以将将背包中的砖块放在第i列。
- 如果第i列有砖块，你可以捡起来，放在背包中。
- 如果第i列和第$i+1$列的高度差少于或等于于k个砖块，你可以从第i列跳到第$i+1$列。

问你是否能从第一列从第1列跳到第n列。
如果能，输出YES,如果不能，输出NO。。

输入格式

多组测试数据

第一行输入组数$T(1 \le T \le 1000)$。

对于每组测试数据

第一行输入$n, m, k(1 \le n \le 100, 0 \le m \le 10^6, 0 \le k \le 10^6)$。

第二行输入n个数，第i个数代表h_i。

输出格式

对于每组数据，输出YES或NO。

原题目

题目描述

Gildong is playing a video game called Block Adventure. In Block Adventure, there are n columns of blocks in a row, and the columns are numbered from 1 to n . All blocks have equal heights. The height of the i -th column is represented as h_i , which is the number of blocks stacked in the i -th column.

Gildong plays the game as a character that can stand only on the top of the columns. At the beginning, the character is standing on the top of the 1 -st column. The goal of the game is to move the character to the top of the n -th column.

The character also has a bag that can hold infinitely many blocks. When the character is on the top of the i -th column, Gildong can take one of the following three actions as many times as he wants:

- if there is at least one block on the column, remove one block from the top of the i -th column and put it in the bag;
- if there is at least one block in the bag, take one block out of the bag and place it on the top of the i -th column;
- if $i < n$ and $|h_i - h_{i+1}| \leq k$, move the character to the top of the $i+1$ -st column. k is a non-negative integer given at the beginning of the game. Note that it is only possible to move to the next column.

In actions of the first two types the character remains in the i -th column, and the value h_i changes.

The character initially has m blocks in the bag. Gildong wants to know if it is possible to win the game. Help Gildong find the answer to his question.

输入格式

Each test contains one or more test cases. The first line contains the number of test cases t ($1 \leq t \leq 1000$). Description of the test cases follows.

The first line of each test case contains three integers n , m , and k ($1 \leq n \leq 100$, $0 \leq m \leq 10^6$, $0 \leq k \leq 10^6$) — the number of columns in the game, the number of blocks in the character's bag at the beginning, and the non-negative integer k described in the statement.

The second line of each test case contains n integers. The i -th integer is h_i ($0 \leq h_i \leq 10^6$), the initial height of the i -th column.

输出格式

For each test case, print "YES" if it is possible to win the game. Otherwise, print "NO".

You can print each letter in any case (upper or lower).

样例 #1

样例输入 #1

```
5
3 0 1
4 3 5
3 1 2
1 4 7
4 10 0
10 20 10 20
2 5 5
0 11
1 9 9
99
```

样例输出 #1

```
YES
NO
YES
NO
YES
```

提示

In the first case, Gildong can take one block from the 1 -st column, move to the 2 -nd column, put the block on the 2 -nd column, then move to the 3 -rd column.

In the second case, Gildong has to put the block in his bag on the 1 -st column to get to the 2 -nd column. But it is impossible to get to the 3 -rd column because $|h_2 - h_3| = 3 > k$ and there is no way to decrease the gap.

In the fifth case, the character is already on the n -th column from the start so the game is won instantly.

思路

每当到了新的一列 h_i，检查下一列的高度 h_{i+1} 是否小于等于 k ，若是则将该列的砖块全部装入背包；否则就检查是否满足 $h_i \geqslant h_{i+1} - k$ ，若满足则把 $i+1$ 列多的砖块取走；若不满足则从背包中取出砖块，尝试使用最少的砖块满足该条件；如果使用完所有砖块后还无法到达下一列，就说明不可能到第 n 列，输出 `NO` ；若成功到达终点，则输出 `YES` 。

```c
#include <cstdio>
#define min(a,b) ((a)<(b)?(a):(b))
int h[106];
int main(void)
{
    int t;
    scanf("%d", &t);
    while (t--) {
        int n, m, k, pd = 1;
        scanf("%d %d %d", &n, &m, &k);
        for (int i = 1; i <= n; i++) {
            scanf("%d", &h[i]);
        }
        for (int i = 1;i <= n - 1; i++) {
            if (h[i+1] <= h[i]) {
                m += min(h[i], h[i] - h[i+1] + k);
            }
            else {
                if (h[i+1] - h[i] > k) {
                    m -= h[i + 1] - h[i] - k;
                    if (m < 0) {
                        pd = 0;
                        break;
                    }
                }
                else if (h[i + 1] - h[i] < k) {
                    m += min(h[i], k - (h[i + 1] - h[i]));
                }
            }
        }
        printf("%s\n", pd ? "YES" : "NO");
    }
    return 0;
}
```