

P5019 铺设道路

题目描述

春春是一名道路工程师，负责铺设一条长度为 n 的道路。

铺设道路的主要工作是填平下陷的地表。整段道路可以看作是 n 块首尾相连的区域，一开始，第 i 块区域下陷的深度为 d_i 。

春春每天可以选择一段连续区间 $[L,R]$ ，填充这段区间中的每块区域，让其下陷深度减少 1。在选择区间时，需要保证，区间内的每块区域在填充前下陷深度均不为 0。

春春希望你能帮他设计一种方案，可以在最短的时间内将整段道路的下陷深度都变为 0。

输入格式

输入文件包含两行，第一行包含一个整数 n ，表示道路的长度。第二行包含 n 个整数，相邻两数间用一个空格隔开，第 i 个整数为 d_i 。

输出格式

输出文件仅包含一个整数，即最少需要多少天才能完成任务。

输入样例

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

输出样例

```
9
```

解析

本题引用到了一个叫做差分概念。在本题中，我们用差分数组来存储，后一个数据与前一个数据之差。那么图1中的原始数组就可以转化为图2中的样子。

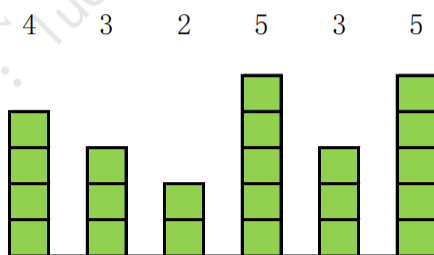


图1

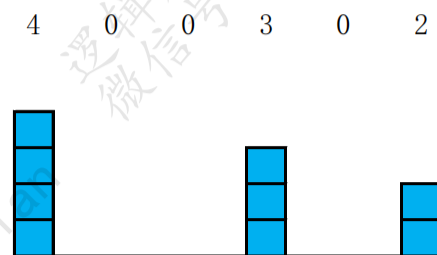


图2

通过上面的两幅图的对比，我们很容易发现规律：

如果后一块的塌陷深度小于前一天，那么它将随着前一天的填充而一齐被填平。

如果后一块的塌陷深度大于前一天，由于每一天的填充都不能为0，因此当前一块被填平的时候，后一块必须放到隔天才能进行填充。

因此，图中的数据最终使用的天数为 $4+3+2 = 9$ 天。

构建差分数组便是本题的关键所在。

编码

```
#include <bits/stdc++.h>

using namespace std;
const int Maxn = 100000 + 1;
int n, countNum = 0;

int Roads[Maxn]; //道路数组
int delta[Maxn]; //差分数组

int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        scanf("%d", &Roads[i]);
        //构建差分数组
        delta[i] = Roads[i] - Roads[i - 1];
        //差分大于0，则进行累加
        //差分数组中的高度就是所需的高度
        if (delta[i] > 0) {
            countNum += delta[i];
        }
    }
    cout << countNum;
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

