

1322: 拦截导弹问题

题目描述

某国为了防御敌国的导弹袭击，开发出一种导弹拦截系统，但是这种拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭，由于该系统还在试用阶段。所以一套系统有可能不能拦截所有的导弹。

输入导弹依次飞来的高度（雷达给出的高度不大于30000的正整数）。计算要拦截所有导弹最小需要配备多少套这种导弹拦截系统。

输入

n颗依次飞来的高度（ $1 \leq n \leq 1000$ ）。

输出

要拦截所有导弹最小配备的系统数k。

输入样例

389 207 155 300 299 170 158 65

输出样例

2

解析

本题的贪心策略是在每次导弹来时都需要从已有系统中寻找能够防御这个导弹的，而且在防御导弹的系统中选择防御度最低的系统，为了将防御高度高的系统留出来。即“以最小损失来完成任务”

对于题目中的数据，我们可以使用如下结构进行存储。

系统1	系统2
389	300
207	299
155	170
65	158

每当一发新的炮弹来临时，我们便从当前已有的系统中找到能够防御该导弹，但是高度又是最低的一个，如果不存在，则需要重新开辟一套系统。

编码

```
#include<bits/stdc++.h>
using namespace std;
//第一维：题目中最多有1000个导弹
//第二维：每个系统中的最小高度
int v[1001][1];
```

//贪心：在现有系统中寻找能够防御的且防御高度最小的系统，将防御高度高的留着

```
int main() {  
    //首先将数组清空  
    memset(v, 0, sizeof(v));  
    //输入的导弹  
    int daodan;  
    //建立的防御系统数量  
    int k = 0;  
    //标记当前这个导弹能否在已有系统中被防御  
    bool flag;  
    //当前导弹被第几个系统所防御  
    int hsys;  
    //循环输入导弹  
    while (cin >> daodan) {  
        //默认当前导弹不可被拦截  
        flag = false;  
        //在现有系统中寻找  
        for (int i = 1; i <= k; i++) {  
            //从左起寻找可以抵御的系统中最小高度的系统，将大限度的系统保留下来  
            //不需要再和右面比较  
            if (v[i][0] >= daodan) {  
                hsys = i; //当前导弹被第几个系统所防御  
                flag = true; //表示这个导弹能被防御  
                break;  
            }  
        }  
        //如果有可以防御这个导弹的，更新系统的最小高度  
        if (flag) {  
            v[hsys][0] = daodan;  
        }  
        //假设没有能防御这个导弹的系统 就设立一个新的系统  
        else if (!flag) {  
            k++;  
            v[k][0] = daodan;  
        }  
    }  
    cout << k;  
    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。

