

1285: 最大上升子序列和

题目描述

一个数的序列 b_i ，当 $b_1 < b_2 < \dots < b_S$ 的时候，我们称这个序列是上升的。对于给定的一个序列 (a_1, a_2, \dots, a_N) ，我们可以得到一些上升的子序列 $(a_{i_1}, a_{i_2}, \dots, a_{i_K})$ ，这里 $1 \leq i_1 < i_2 < \dots < i_K \leq N$ 。比如，对于序列 $(1, 7, 3, 5, 9, 4, 8)$ ，有它的一些上升子序列，如 $(1, 7)$, $(3, 4, 8)$ 等等。这些子序列中和最大为18，为子序列 $(1, 3, 5, 9)$ 的和。

你的任务，就是对于给定的序列，求出最大上升子序列和。注意，最长的上升子序列的和不一定是最大的，比如序列 $(100, 1, 2, 3)$ 的最大上升子序列和为100，而最长上升子序列为 $(1, 2, 3)$ 。

输入

输入的第一行是序列的长度 N ($1 \leq N \leq 1000$)。第二行给出序列中的 N 个整数，这些整数的取值范围都在0到10000(可能重复)。

输出

最大上升子序列和。

输入样例

```
7
1 7 3 5 9 4 8
```

输出样例

```
18
```

解析

本题其实很简单，我们之前求最长子序列时，是使用的一个dp数组存储最长子序列的长度，那么在本题当中，这个dp数组就不再需要存储长度，而改为存储最大值即可。

编码

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

using namespace std;

int a[1001];      //原始数组
int dp[1001];    //最大和数组
int n;           //数组的长度
```

```

int main(int argc, char **argv) {
    //读入数据
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        //将最值赋值为初始数据
        dp[i] = a[i];
    }
    int m = 0;
    //模拟上升子序列计算过程
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {
            //求上升子序列的和
            if (a[j] < a[i]) {
                //最值比较
                dp[i] = max(dp[j] + a[i], dp[i]);
            }
        }
        //全局的最值比较
        m = max(dp[i], m);
    }
    cout << m;
    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。

