

1259: 求最长不下降序列

题目描述

设有由 n ($1 \leq n \leq 200$) 个不相同的整数组成的数列, 记为: $b(1)$ 、 $b(2)$ 、……、 $b(n)$ 若存在 $i_1 < i_2 < i_3 < \dots < i_e$ 且有 $b(i_1) \leq b(i_2) \leq \dots \leq b(i_e)$ 则称为长度为 e 的不下降序列。程序要求, 当原数列出之后, 求出最长的不下降序列。

例如13, 7, 9, 16, 38, 24, 37, 18, 44, 19, 21, 22, 63, 15。例中13, 16, 18, 19, 21, 22, 63就是一个长度为7的不下降序列, 同时也有7, 9, 16, 18, 19, 21, 22, 63组成的长度为8的不下降序列。

输入

第一行为 n , 第二行为用空格隔开的 n 个整数。

输出

第一行为输出最大个数 \max

(形式见样例);

第二行为 \max 个整数形成的不下降序列, 答案可能不唯一, 输出一种就可以了, 本题进行特殊评测。

输入样例

```
14
13 7 9 16 38 24 37 18 44 19 21 22 63 15
```

输出样例

```
max=8
7 9 16 18 19 21 22 63
```

解析

最长不下降序列包含相等的情况

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

using namespace std;
const int maxn = 201; //序列长度最长为200
int a[maxn], f[maxn], c[maxn];
int k, n, ans = -INT_MAX; //防止整个数组为降序数组
int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        scanf("%d", &a[i]);
        f[i] = 1;
    }
    //遍历数组中的每一个数字
    for (int i = 1; i <= n; i++) {
        //开始向前寻找
        for (int j = 1; j < i; j++) {
            //找到第一个比他小的数字
            if (a[j] <= a[i]) {
                //比较前一个比他小的数字加1后的长度和自己当前的长度谁更大
                //需要比较是因为可能会出现波谷数，例如 1 3 4 1 4，
                //当第二个4遇到第二个1的时候，此时f(4)的长度为3，f(1)仅为1
                f[i] = max(f[i], f[j] + 1);
            }
            if (f[i] > ans) {
                ans = f[i];
                k = i; //当前拥有最大LIS的索引
            }
        }
    }
    printf("max=%d\n", ans);

    int t = 0;
    int m = ans; //最长的LIS的数值
    int i = k;
    //c[]数组存储最长不下降序列的每一个下标，m为最长的不下降序列的总数字和，
    // 每次循环完就要-1，i为所有数字下标
    while (m >= 1) {
        //1、找到LIS符合需求的
        //2、数据小于a[k]
        if (f[i] == m && a[i] <= a[k]) {
            //maxIndex下标记上一个数字，i为下标为此时的数字
            c[t++] = i;
            k = i;
            m--;
        }
    }
}
```

```
    }  
    //不断的向前寻找  
    i--;  
}  
  
for (i = t - 1; i >= 0; i--) {  
    printf("%d ", a[c[i]]);  
}  
  
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。

