

1259 : 【例9.3】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： $b(1)$ 、 $b(2)$ 、……、 $b(n)$ 且 $b(i) \neq b(j)(i \neq j)$ ，若存在 $i_1 < i_2 < i_3 < \dots < i_e$ 且有 $b(i_1) < b(i_2) < \dots < b(i_e)$ 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 【来源】

http://ybt.ssoier.cn:8088/problem_show.php?pid=1259

思路

对求个数的说明

当数列规模是1个数的时候，如13，那么 $mm[1]=1$

当数列规模是2个数的时候，如13 7，那么 $mm[1]=1$ $mm[2]=1$

当数列规模是3个数的时候，如13 7 9，那么 $mm[1]=1$ $mm[2]=1$ $mm[3]=2$ 因为9比7大，所以以9结尾的不降子序列，是7 9

当数列规模是4个数的时候，如13 7 9 16，那么 $mm[1]=1$ $mm[2]=1$ $mm[3]=2$ $mm[4]=3$ 因为16比9大，所以以16结尾的不降子序列，是7 9 16 这种方案要比7 16这样的子序列要好

当数列规模是5个数的时候，如13 7 9 16 38，那么 $mm[1]=1$ $mm[2]=1$ $mm[3]=2$ $mm[4]=3$ $mm[5]=4$

当数列规模是6个数的时候，如13 7 9 16 38 24，那么 $mm[1]=1$ $mm[2]=1$ $mm[3]=2$ $mm[4]=3$ $mm[5]=4$ $mm[6]=4$ ，以24结尾的最长不降子序列是7 9 16 24

。。。以此类推

对递归输出的说明

以样例为例 `print (13)` 将会输出 答案 7 9 16 18 19 21 22 63

将这个问题的规模缩小 `print (prd[13])` ,也就是`print(12)`，应该会输出7 9 16 18 19 21 22

所以`print(n)`这个大问题，分解后变成`print(n-1)`,然后在输出`a[n]`

最后在确定基条件：`if(x==0) return;`

源代码

```
1 #include<iostream>
2 using namespace std;
3 int n,a[300],mm[300],pre[300],maxx,maxid;
4
5 void deal(){
```

```

6     mm[1]=1;
7     for(int i=2;i<=n;i++){
8         mm[i]=1;
9         for(int j=i-1;j>=1;j--){
10             if(a[j]<=a[i] && mm[j]+1>mm[i]){
11                 mm[i]=mm[j]+1;
12                 pre[i]=j;
13             }
14         }
15     }
16 }
17 void print(int x){
18     if(x==0) return;
19     print(pre[x]);
20     cout<<a[x];
21     if(x!=maxid) cout<<' ';
22 }
23 int main(){
24     cin>>n;
25     for(int i=1;i<=n;i++){
26         cin>>a[i];
27     }
28     maxx=mm[1];
29     maxid=1;
30     for(int i=2;i<=n;i++){
31         if(maxx<mm[i]) {
32             maxx=mm[i];
33             maxid=i;
34         }
35     }
36     cout<<"max="<<maxx<<endl;
37     print(maxid);
38     return 0;

```