

Wavio Sequence

题目唯一标示:180105

考察内容:

最长上升子序列

题目分析:

给出一个序列，从中找出尽可能多的单数个数，使得前一半递增，后一半递减。

考虑分别正序、倒叙求出最长上升序列，设正序为up[]，倒叙为down[]，up/down[i]：前i个数字能组成的最长上升/下降序列。则求出 $\max(\min(\text{up}[i], \text{down}[i]))-1$ ，即是答案。因为你可以找到一个长度为 $\min(\text{up}[i], \text{down}[i])$ 且最后一个数是i的最长上升序列，也可以找到一个长度相同且第一个数是i的最长下降序列。

下面是代码：（emmm，还是彩色的）

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 05 12
*文件类型: 源代码文件
*题目来源: UVA
*当前状态: 已通过
*备忘录: 动态规划 DP 最长上升子序列 LIS
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
using namespace std;
#define MAXN 10005
int n, a[MAXN], f1[MAXN], f2[MAXN], up[MAXN], down[MAXN];
/* Variable explain:

*/
void read()
{
    if(scanf("%d",&n)==-1)exit(0);
    for(int i=1;i<=n;++i)
        scanf("%d",a+i);
    return;
}
int find(int x,int n,int f[])
{
    int maxa=n,mina=1,ans=0;
    while(maxa>=mina)
    {
        int mid=(maxa+mina)>>1;
        // printf("check:%d>%d\n",x,f1[mid]);
        if(x>f[mid])ans=mid,mina=mid+1;
        else maxa=mid-1;
    }
}
```

```

    }
    return ans+1;
}
void dp()
{
    int ans=0;
    for(int i=1;i<=n;++i)
    {
        int x=find(a[i],ans,f1);
        f1[x]=a[i];
        if(x==ans+1)++ans;
        up[i]=ans;
    }
    ans=0;
    for(int i=n;i>=1;--i)
    {
        int x=find(a[i],ans,f2);
        f2[x]=a[i];

        if(x==ans+1)++ans;
        down[i]=ans;
    }
    ans=0;
    for(int i=1;i<=n;++i)
        ans=max(ans,2*min(up[i],down[i])-1);
    printf("%d\n",ans);
}
int main()
{
    while(1)
    {
        read();
        dp();
    }
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)