

## 20160202考题 array(单调栈好题)

题意：

相传，瑞士数学家、自然科学家莱昂哈德·欧拉也研究过一种奇妙的有穷数列A，它是由另一个有穷数列B导出的。数列A性质奇特，虽然每一项与相邻几项并无直接关系，但是若将数列A整体看来，却与B有着宏观上的近似之处。它具体表现为 $a_n = n - k + 1$ 其中k为最小的正整数使得数列B中对于所有 $k \leq i \leq n$ 的i均满足 $b_k \leq b_i \leq b_n$ 。我们并不关心欧拉是如何逐项计算A的，但我们颇感兴趣的题是A中的最大值是多少。

题解：

- 1.对于每一个位置，比较裸的一个思路是找到以它为右端点且为最大值的最长区间内的最小值的最左端的位置即为它的k值。
- 2.我们把每个右端点求出来的k叫做他的左端点的话，可以看成一些区间，然后发现这些区间并没有相交的地方，假设相交一定会有延伸。
- 3.而且这些k是可以用前面的k来更新后面的k的，仔细根据左右端点大小关系发现是可以的。
- 4.因此得到一个优越的 $O(n)$ 算法，即维护一个以 $b_i$ 为键值单调不上升栈，顺便存一个k值，然后现在拿到了一个数，我们可以把所有小于等于他的全部弹掉并且更新他的答案（因为大于他的已经不是最小值了），考虑怎么更新答案。
- 5.因为之前的区间有可能是有包含关系的，所以不能够盲目的只用最小的那个来更新（大概是因为我比较蠢之前就是这样想的），而是要对栈内k值所在位置的数与当前我的k所在的数比大小，只有更小才行，然后k就被妥妥的更新啦。
- 6.至此题目已经做完了一大半，然后要手写栈，stl被卡的人伤心的哭出来。

```
#include<cstdio>
#include<iostream>
#include<cctype>
using namespace std;
const int maxn=10000000+5;
int a[maxn];
```

```
struct node{
    int val,k;
}sta[maxn];
int top;
void read(int &x){
    int f=1,y=0;
    char c=getchar();
    while(isspace(c)) c=getchar();
    if(c=='-'){
        f=-1;
        c=getchar();
    }
    while(c<='9'&&c>='0'){
        y=10*y+c-'0';
        c=getchar();
    }
    x=f*y;
}
int main(){
    freopen("array.in","r",stdin);
    freopen("array.out","w",stdout);
    int n;
    int ans=0;
    scanf("%d",&n);
    for(int i=1;i<=n;++i){
        read(a[i]);
        int k=i;
        while(top&&sta[top].val<=a[i]){
            node tmp=sta[top];
            if(a[tmp.k]<=a[k]&&tmp.k<k) k=tmp.k;
            top--;
        }
        sta[++top]=(node){a[i],k};
        ans=max(ans,i-k+1);
    }
    printf("%d\n",ans);
    return 0;
}
```

2016-02-02