

# 单调栈

x

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

using namespace std;

int main() {

    stack<int>s;//从栈底到栈顶单调递增的单调栈 （出栈为递减顺序）

    int n;//向右遍历查找第一个比它小的元素 用底到顶递增的栈。

    n=4;

    int a[5]={0,5,3,7,4};//下标从1到n

    int l[5]);// l[i]保存的是从a[i]开始向右遍历第一个比a[i]小的元素的位置. 如果为0则表示没有.

    memset(l,0,sizeof(l));

    for(int i=n;i>=1;i--)

    {

        while(s.size()&&a[s.top()]>=a[i]) s.pop();//当栈不为空且栈顶元素大于等于当前元素

        if(s.empty()) l[i]=0;//如果为空表示向右遍历没有比它小的数

        else l[i]=s.top();//否则s.top()为第一个比a[i]小的元素的位置

        s.push(i);

    }

    // 模拟入栈过程：

    //      i=4 a[4]=4 栈为空 l[4]=0 下标4入栈 （4）

    //      i=3 a[3]=7比栈顶下标对应元素大 l[3]=4 下标 3入栈 （3,4）

    //      i=2 a[2]=3 3出栈，4出栈 l[2]=0 2入栈

    //      i=1 a[1]=5 l[1]=2, 1入栈

    for(int i=4;i>=1;i--)//0 4 0 2

        cout<<l[i]<<endl;

    while(!s.empty())//栈中从底到顶为 2 1

    {

        cout<<s.top()<<endl;

        s.pop();

    }

}
```