

单调栈

主讲老师：党东





一、单调栈



【单调栈的性质】



- **单调栈**是指一个**栈内部的元素具有严格单调性**的一种数据结构，分为单调递增栈和单调递减栈。

其具有以下两个性质：

- 满足栈底到栈顶的元素具有严格单调性。
- 满足栈的先进后出特性，越靠近栈底的元素越后出栈。

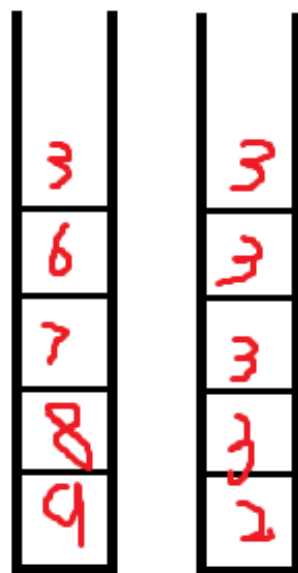


图1

图2

如图所示：

图1所示栈为单调递减栈。

图2所示栈既不是单调递减栈，也不是单调递增栈。

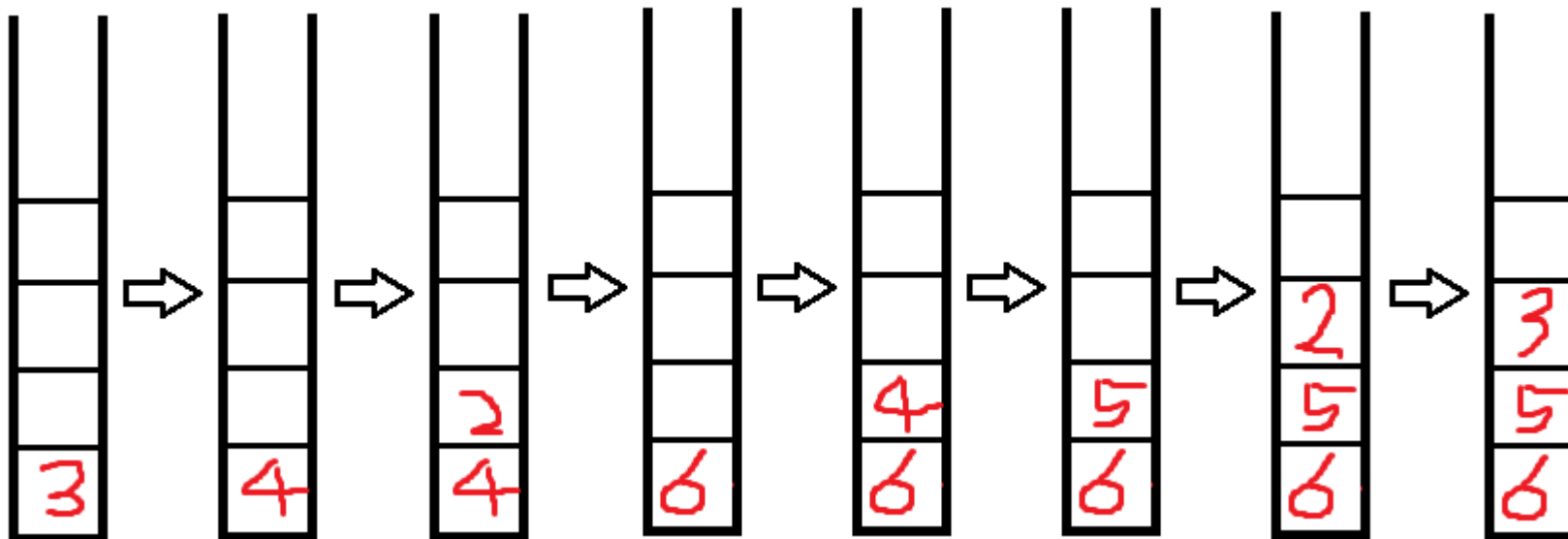


【元素的进栈过程】



■ 元素进栈过程：

- 对于一个单调递增栈来说，若当前进栈的元素为 a ，如果 $a < \text{栈顶元素}$ ，则直接将 a 进栈。
 - 如果 $a \geq \text{栈顶元素}$ ，则不断将栈顶元素出栈，直到满足 $a < \text{栈顶元素}$ 。
- 模拟一个数列构造一个单调递增栈，进栈元素分别为3, 4, 2, 6, 4, 5, 2, 3。
- 图片所示过程即为进栈过程。



实现单调栈
STL栈和手写栈均可

【例1】统计 --1545



Description

现给你 n ($n \leq 10^6$) 个整数，要求你找出每个数后面第一个比它大的数的编号。若没有，则输出0。

Input

第一行一个整数 n ；第二行有 n 个由空格分开的整数。

Output

输出文件仅一行为 n 个由空格分开的整数。分别表示每个数后面第一个比它大的数的编号。若没有，则输出0。

Sample Input

```
6
3 2 6 1 1 2
```

Sample Output

```
3 3 0 6 6 0
```

【数据范围】 对于100%的数据， $n \leq 10^6$ ；

【例1】统计 --1545



```
#include<stdio>
int f[1000005],s[1000005],a[1000005],top,x,n;
//单调栈s维护编号
int main(){
    scanf("%d",&n);
    for(int i=1;i<=n;i++) scanf("%d",&a[i]);
    for(int i=n;i>=1;i--){//单调入栈
        while(top&&a[s[top]]<=a[i]) top--;
        f[i]=s[top];
        s[++top]=i;
    }
    for(int i=1;i<=n;i++) printf("%d ",f[i]);
    return 0;
}
```

【例2】矩形覆盖 --1546

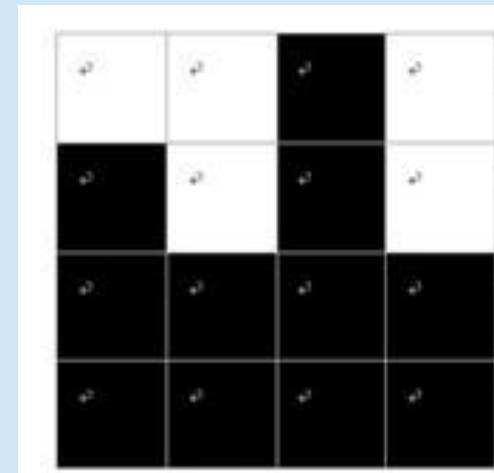


Description

有N个矩形，矩形的底边边长为1，且均在X轴上，高度给出，第i个矩形的高为 $h[i]$ ，例如 $h=[3,2,4,2]$ 的图形如下：

你可以容易地发现，只需要3个矩形就能覆盖这个图形。

你的任务就是，输出最少需要几个矩形能覆盖这个图形。



Input

第一行一个整数N。接下来1行包含N个正整数，为 $h[i]$ 。

Output

输出一个整数表示最少需要几个矩形能覆盖这个图形。

Sample Input

```
10
2 3 2 4 2 1 3 4 3 2
```

Sample Output

```
7
```

【数据规模】对于所有数据， $N \leq 100000$ ， $h[i] \leq 100$ 。

【例2】矩形覆盖 --1546



```
int f[1000005],top,x,n,ans;
int read(){
    char c;int x;while(c=getchar(),c<'0' || c>'9');x=c-'0';
    while(c=getchar(),c>='0' && c<='9') x=x*10+c-'0';
    return x;
}
int main(){
    n=read();top=0;f[0]=-1;
    for(int i=1;i<=n;i++){
        x=read();
        while(x<f[top]) top--;
        if(x==f[top]) continue;
        if(x>f[top]) ans++, f[++top]=x;
    }
    printf("%d",ans);
    return 0;
}
```


【练习】距离最近的人 --1543



Description

有 n 位贵宾排成一排等待晚餐。每个贵宾需要找出前面与自己最靠近且比自己矮的那位，然后与他（她）共进晚餐，由于这些贵宾的身份都很高贵，不愿意屈尊亲自去寻找，这样可难坏了主人John。为了满足每位贵宾那可怜的虚荣心，使晚餐能有序而顺利地进行，需要你来为John来排忧解难。

Input

数据的第一行是一个正整数 n ，表示一共有 n 个人。

第二行有 n 个用空格隔开的正整数，它们从左至右给出了 n 个人的身高。这些数保证小于 2^{31} 。

Output

n 个用空格分开的数，第 i 个数表示比第 i 个人身高矮的与他最近哪个人的身高，如果没有符合条件的，则输出0。

Sample Input

7

3 1 2 7 6 7 4

Sample Output

0 0 1 2 2 6 2

【数据规模】 对于100%的数据， $n \leq 500\ 000$ 。

【练习】距离最近的人 --1543



```
#include<iostream>
#include<cstdio>
using namespace std;
int a[500005],top,i,x,n;
int main()
{ a[0]=0;top=0;
  scanf("%d",&n);
  for(i=1;i<=n;i++)//对于n个人依次处理
  { scanf("%d",&x);
    while(top>0&&a[top]>=x)top--;//维护栈满足递增关系
    printf("%d ",a[top]);
    a[++top]=x; //入栈
  }
  return 0;
}
```