# 单调栈及应用

## 一：单调栈

### 递增栈：

### 出栈的序列是递增的。如栈低到栈顶：10，9，8，7，6。则出栈序列为6，7，8，9，10

### 递减栈：

### 出栈的序列是递增的

用于求区间，非常使用。

# 二：模板伪代码

## 递增栈的维护.

(递减栈就把下方的改为<号: 当前数组元素 < 栈顶元素)

for(遍历数组) {

while(栈空 && 当前数组元素 > 栈顶元素) {

出栈

更新结果,一般会利用坐标差

}

当前数组元素入栈

}

# 三：应用

## 2.视野总和

描叙：有n个人站队，所有的人全部向右看，个子高的可以看到个子低的发型，给出每个人的身高，问所有人能看到其他人发现总和是多少。  
输入：4 3 7 1  
输出：2  
解释：个子为4的可以看到个子为3的发型，个子为7可以看到个子为1的身高，所以1+1=2。

思路如下：即求每一个元素到其右边第一个比他大的元素的区间，然后相加。

int vis\_sum(int\* num, int size)

{

int\* stack = (int\*)malloc(sizeof(int) \* (size));

int\* v = (int\*)malloc(sizeof(int) \* (size + 1));

int sum = 0;

int i = 0, top = -1, temp = 0;

memset(stack, 0, sizeof(int) \* (size));

memset(v, 0, sizeof(int) \* (size + 1));

memcpy(v, num , sizeof(int) \* (size));

v[size] = 100;/ /理解为在最后插入一个最高的人，否则无法全部将有效数据弹出弹出

for (i = 0; i < size +1; i++) {

printf("%d ",v[i]);

}

for (i = 0; i < size + 1; i++) {

while (top != -1 && num[i] > num[stack[top]]) {

temp = stack[top];

top--;

sum += (i - temp - 1); }

top++;

stack[top] = i;

}

return sum;

}

int main()

{

int num[] = {4, 3, 7, 1};

printf("\n%d\n",vis\_sum(num, 4));

}