二. ST表

区间倍增

yt1541

【题目描述】

输入一串数字，给你 M 个询问，每次询问就给你两个数字 X,Y，要求你说出 X 到 Y 这段区间内的最大数。

【输入】

第一行两个整数 N,M 表示数字的个数和要询问的次数；

接下来一行为 N 个数；

接下来 M 行，每行都有两个整数 X,Y。

【输出】

输出共 M 行，每行输出一个数。

【输入样例】

```
10 2
3 2 4 5 6 8 1 2 9 7
1 4
3 8
```

【输出样例】

```
5
8
```

【提示】

数据范围与提示：

对于全部数据，$1 \le N \le 10^5, 1 \le M \le 10^6, 1 \le X \le Y \le N$。数字不超过 $C/C++$ 的 int 范围。

RMQ(Range Minimum/Maximum Query)问题

- 指求区间的最值问题：
  对于长度为n的数列A，回答Q个询问RMQ(A,i,j)(i,j<=n)，返回数列A中下标在[i,j]里的最小(大)值。

  动态查询数组元素A[L],A[L+1],…,A[R]中的最大值（最小值）问题。
  条件：数组元素保持不变。

  N=50,000
  Q=200,000

1.朴素算法：

- 直接枚举，每个询问的时间是O(n)
- q个查询的时间是O(q*n)。

2.线段树

- 用线段树来记录这个最大（小）值，预处理（建树）时间是O(n)，每个区间查询的复杂度都是O（log(n)）.
- 总的时间复杂度O(n+Q*log(n))

3. 算法：ST算法：

- ST（Sparse Table：稀疏表）算法
- 它可以做到O(nlogn)的预处理
- O(1)地回答每个询问。
- 总的时间复杂度O(nlogn+Q)
- 区间倍增，代码简单

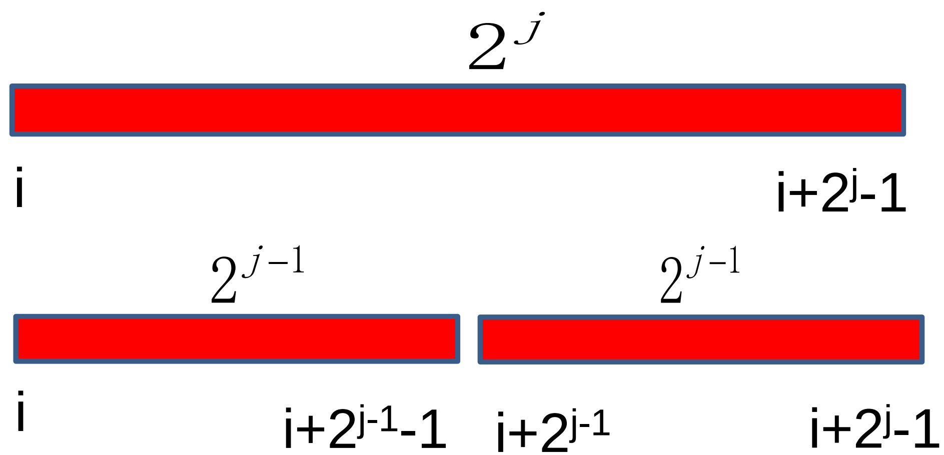
倍增思想：

- 设f[i][j]表示从第i个数起连续 2^j个数中的最大值。
- F[i][0]表示第i个位置开始，往后1个数的最大值是多少
- F[i][1]表示第i个位置开始，往后2个数的最大值是多少
- F[i][2]表示第i个位置开始，往后4个数的最大值是多少
- F[i][3]表示第i个位置开始，往后8个数的最大值是多少
- ············
- 最多LogN行

◆例如数列a[10]

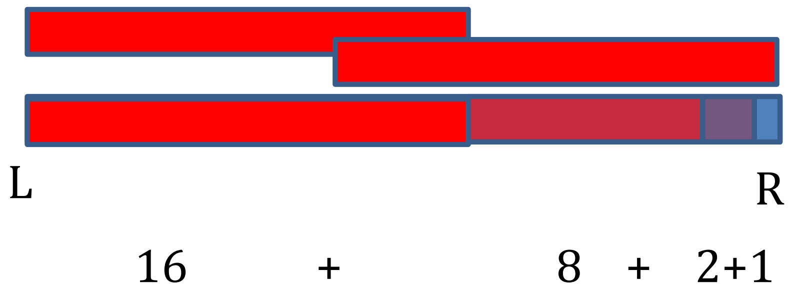
1	2	3	4	5	6	7	8	9	10
3	2	4	5	6	8	1	2	9	7

设f[i][j]表示从a[i]开始连续2^j个元素中的最大值。
显然有f[i][0]=a[i]作为初始的边界状态。



$$f[i][j] = \max(f[i][j-1], f[i+2^{j-1}][j-1])$$

求区间[L，R]中的最大值,[10,36]



等长16的两段即可！

对于任意的L~R，一定能拆分成不超过Log(R-L+1)段

根据2k<=R-L+1,得到最大的k：　　K=(log(R-L+1)/log(2))

[L,L+2^k-1]和[R-2^k+1,R]

$Q(L,R)=\max(f[L][k],f[R-2^k+1][k])$

◆ST算法的实现：

- 第一步：生成st表，简单的过程DP，求f[i][j]

```
void ST(){
    int k=log(n)/log(2);
    for(int i=1;i<=n;i++)f[i][0]=a[i];//初始化
    for(int j=1;j<=k;j++)  //注意枚举顺序区间从小到大
        for(int i=1;i+(1<<j)-1<=n;i++)
            f[i][j]=max(f[i][j-1],f[i+(1<<(j-1))][j-1]);
}
```

注意区间范围：长度为2^j的区间：$[i,i+2^j-1]$
不能超过$[1,n]$

第二步：查询区间[L,R]中的最大值

```
int RMQ(int L,int R){
    int k=log(R-L+1)/log(2);
    return max(f[L][k],f[R-(1<<k)+1][k]);
}
```

◆1. 【例 1】数列区间最大值yt1541

2：cow(poj3264)

- 【题目描述】
- 有一排n头牛，给你每个牛的高度h，然后有q个询问，即给你i，j要你求第i个牛和第j个牛之间的最高的牛和最矮的牛之间的差值是多少？
- 【输入】
- 第一行：n，q。n是牛的个数；q询问数。
- 第二行：n个数。依次表示n头牛的高度*height*（1 ≤ *height* ≤ 1,000,000）。
- 以下q行询问，每行i，j。1<=i<=j<=n。
- 【输出】
- q行，依次为q个询问的结果。

- 【样例输入输出】
- cow.in
- 6 3
- 1 7 3 4 2 5
- 1 5
- 4 6
- 2 2
- cow.out
- 6
- 3
- 0

3. Frequent values（poj3368） 测试T2

- 给定一个数组，其中的元素满足非递减顺序，要求对于一对起点和终点，回答出其中某个元素重复出现的最多次数。
- 比如对于-1 -1 1 1 1 1 3 10 10 10，若起点为1，终点为5，则重复出现最多的数是1，其次数为3。
- **输入：**
- n ，q （$1 \leqslant n, q \leqslant 100000$）；
- a_1 ，... ，a_n （$-100000 \leqslant a_i \leqslant 100000$）；$a_i \leqslant a_{i+1}$；
- *以下q行：* i and j （$1 \leqslant i \leqslant j \leqslant n$）。
- **输出：**
- 每个询问区间的最多次数。

- Sample Input
- 10 3
- -1 -1 1 1 1 1 3 10 10 10
- 2 3
- 1 10
- 5 10
- Sample Output
- 1
- 4
- 3

-1 -1 1 1 1 1 3 10 10 10

- 原数组是非递减的，因此相同的数字必然是连续出现的。可以对原数组进行压缩，把相同数字构成的区间抽象为一个数，这个数就是这个区间的大小。
- 比如上述数组可以压缩为:2 4 1 3。
- 对于原数组中的每个位置，记录它属于第几个区间，并记录每个区间的起点和终点。
  对于一对起点i和终点j，首先计算出它们各属于哪个区间，分为三种情况:
  1. 属于同一个区间：答案就是它们之间的大小
  2. 属于相邻区间：找到它们的分隔点，取两个区间大小的较大者
  3. 属于两个不相邻的区间：取出头尾两个区间，计算它们的大小。对于中间的一个或多个区间，它们都是完整的，因此可以使用求解RMQ问题ST算法，计算出它们中的最大值。最后的答案即为该最大值与头尾两个区间的大小，一共三个数的最大值。

i	1	2	3	4	5	6	7	8	9	10
a[i]	-1	-1	1	1	1	1	3	10	10	10

压缩后区间编号	1	2	3	4
次数值	2	4	1	3
区间范围	[1,2]	[3,6]	[7,7]	[8,10]

p[i]:a[i]所在的区间编号：

p[1]=p[2]=1;

p[3]=p[4]=p[5]=p[6]=2;

p[7]=3;

p[8]=p[9]=p[10]=4;

s[i],t[i]:压缩后区间i在原始区间的起点和终点：

差就是次数t[i]-s[i]+1

s[1]=1;t[1]=2

s[2]=3;t[3]=6

s[3]=7;t[3]=7

s[4]=8;t[4]=10

- 询问区间L,R:
- 求出L和R所在的区间编号：x=p[L],y=p[R]
- x=y：[L,R]=[3,5] 属同一个区间：ans=R-L+1;
- y-x=1:[L,R] =[2,5]属于相邻区间：找到它们的分隔点，取两个区间大小的较大者ans=max(t[x]-L+1,R-s[y]+1)
- y-x>1:[L,R]=[2,9] 属于两个不相邻的区间：取出头尾两个区间，计算它们的大小。对于中间的一个或多个区间，它们都是完整的，因此可以使用求解RMQ问题ST算法，计算出它们中的最大值。最后的答案即为该最大值与头尾两个区间的大小，一共三个数的最大值。
- ans=max(t[x]-L+1,RMQ(x+1,y-1),R-s[y]+1)
- 其中：f[1][0]=2，f[2][0]=4,f[3][0]=1,f[4][0]=3