## 概要

对于一个数列a1,a2,a3……每次询问max(i,j)或者min(i,j)，即ai,a(i+1)……aj这一区间中的最大值或最小值。

ST算法可以先用O(nlogn)的时间进行预处理，然后之后只用O(1)的时间就能给出答案。

原理是可以把任意子序列划分成2^i的几段，使用dp[i][j]表示以i为起始下标，长为2^j的子段（注意，除了长为1的情况，子段长度都是偶数）中的最值，假设数列存于数组x[]中，那么显然dp[i][0]=x[i]，这也是动态规划预处理时的边界条件。

J!=0，时，可以把dp[i][j]划分成长度相等的两段dp[i,j-1]，dp[i+2^(j-1)，j-1]，即i到i+2^(j-1)-1为一段，i+2^(j-1)到i+2^j-1，则求i到j的最值，显然就是求两个子段分别的最值的最值。状态转移方程：

dp[i,j]=max/min（dp[i，j-1],dp[i+2^(j-1),j-1]）

预处理时，对所有的i，先计算dp[i][1]然后计算dp[i][2]……注意好别越界。假设数组长为n，下标从1开始，则j的范围为0到(int)log2(n)。

求解时，因为所求区间[i,j]长度(j-1+1)不一定是2的幂，所以可以取k=log2(j-i+1)，显然2^k<=区间长度，于是可以把区间分成左半部分和右半部分，长度都为2^k，即使有重叠也没关系的，然后求这两个区间分别的最值的最值，就是最值要求的最值。

## 区间出现次数最多的值的出现次数

一个不严格递增的数列，每次询问任意区间内出现次数最多的值的出现次数。

因为数列是不严格递增的，所以相同的值肯定是聚在一起的，可以将数列分成许多段，每一段内元素值都相同，那么对于询问的区间，可能只包含了1段（此时答案就是区间长度），也可能包含了2段及以上，此时答案就是以下3种段中的最大值：

1. 左端点，至所在段末尾
2. 右端点，至所在段起始
3. 左端点与右端点所在段，中间那些段（若存在）中最长的段

所以需要记录数列中每一个元素对应的段号，每个段号对应段的段长，段的

起始、终点。在段长数组上使用ST算法。

## 区间内任意两值差的最大值

利用ST算法求出求出区间内最大值与最小值，相减就行了。

## 二维RMQ

一个N\*N的矩阵，每次询问以(i,j)为左上角的P\*P的子矩阵中最大值与最小值的差，保证子矩阵在原矩阵范围内。

没去管二维RMQ的模板，第一时间想到的就是把二维矩阵按一维存，在每一行上使用ST算法进行预处理，询问时也在子矩阵的每一行中查，最后再取最值。稍微改下一维RMQ的模板就行了。