序列3

栈、队列

- 先入后出、先入先出

单调栈、单调队列

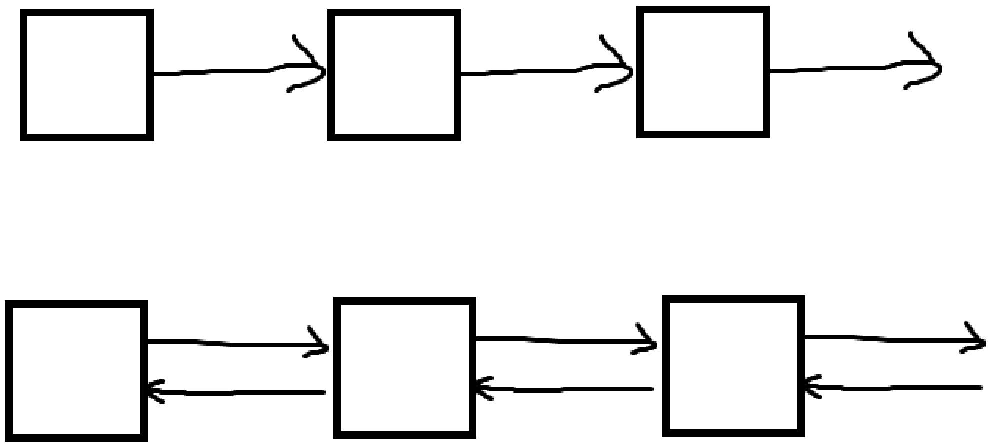
- 元素单调的栈。
- 元素单调的队列。

- 对所有元素求左侧最近的比自己大的元素?
- 若 $i < j$ 且 $a_i \le a_j$ ，则对于在 j 右侧的元素， j 一定比 i 更优。
- 所以有可能成为答案的元素一定满足：
- $i < j < \cdots < k$ ， $a_i > a_j > a_k$ 。
- 加入元素ai?
- while(top&&stk[top]<=ai)top--;
- 此时stk[top]就是ai左侧第一个比ai大的数。
- stk[++top]=ai;

滑动窗口最大值

- 单调队列
- 调整队首元素。

链表



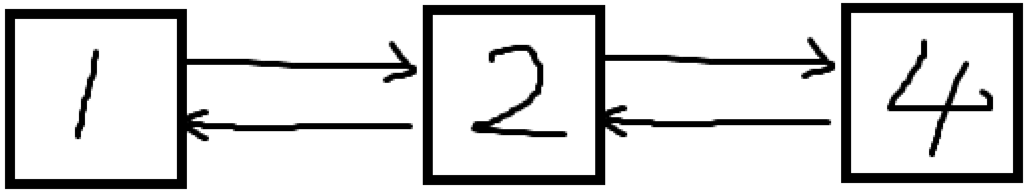
```c
struct List
{
    struct List *prev;
    struct List *next;
    //Data
};

struct List
{
    int prev;
    int next;
    //Data
}list[N];
```

- 向一个集合中插入n个数，每次插入时求出当前集合中距离这个数最近的数。

- 3
- 4 1 2

- -1 4 1

- 在线：set
- 离线：排序后双向链表，从后往前删除元素。



- bzoj1588

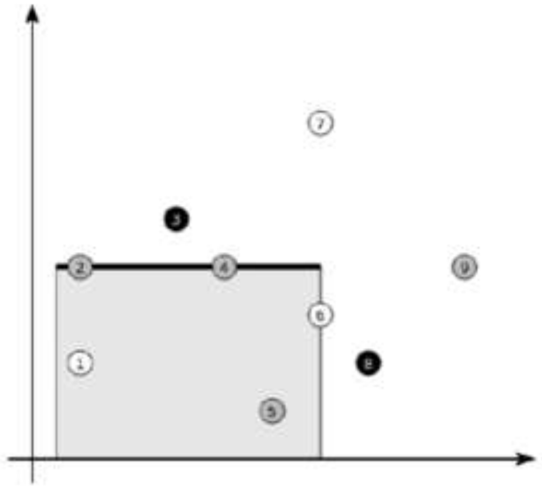
双指针

- 对于每个左端点，满足条件的最大右端点单调不降。

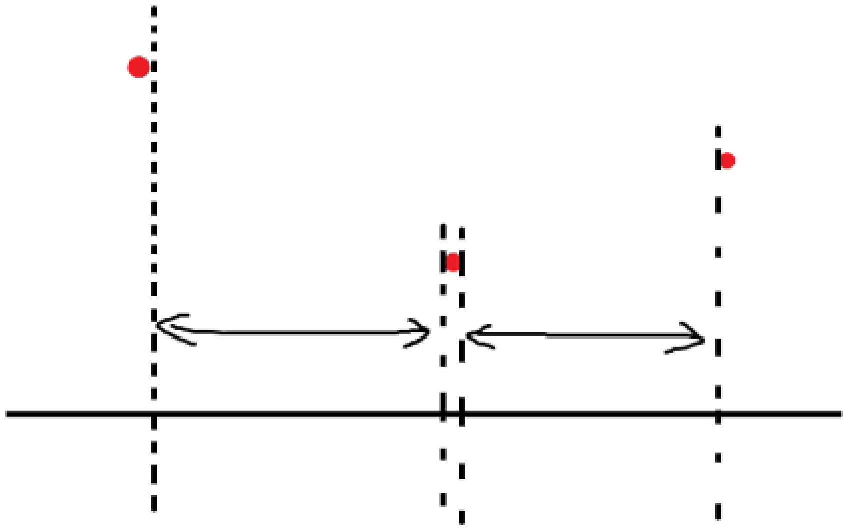
状态压缩DP

- 5:101

- 平面上有n个点，每个点有k种颜色中的一个。你可以选择一条水平的线段获得在其上方或其下方的所有点，请求出你最多能够得到多少点，使得获得的点并不包含所有的颜色。



- 某种颜色不取。
- 固定线段的横坐标?
- 某种颜色纵坐标相邻两点。



- 从下往上枚举横坐标。
- 每次删去一个点。
- 找到前驱后继，统计区间内的点数。
- 双向链表+BIT。

- bzoj3658

- n 个物品， m 个金币。
- 对于每个物品有三种选择：
- 1.不花费金币。
- 2.花费一枚金币，获得价值 a_i 。
- 3.花费两枚金币，获得价值 b_i 。
- 求能获得的最大价值。
- $n \leq 10^5$
- 5 6
- 2 4
- 3 8
- 3 9
- 9 10
- 1 9

16

22101

- 按 B_i 排序，枚举最后一个花费为二的位置。
- 之前的至少花费一，之后的至多花费一。
- 维护前 k 大之和。

- $O(nlogn)$

- 长为n的序列，q次操作。
- 第 i 次操作，要么对一个区间 $[l_i, r_i]$ 加上首项为 a_i ，公差为 b_i 的等差数列，要么询问区间 $[l_i, r_i]$ 最少能划分成几个等差数列。

- 1 3 -1 -4 7
- add [2,4],a=-1,b=5
- query [1,5]

- 1 2 3 5 7

- {1,2,3},{5,7}或{1,2},{3,5,7}

- 差分，区间加等差数列转化为端点处的单点修改和区间和。
- 询问?
- 在区间中删除最少的数字，剩余的每一段都相等。
- 1 2 3 4
- 1 3 6 10
- [1,3],[6,10]
- 线段树，每个节点存左右端点值、区间内段数、左右长度为1的连续段数量。
- bzoj1558

- 长为n的序列a，取出m个元素，要求相邻的不能一起取（首尾也算相邻），求取出数的最大和。

- 7
- 1 2 3 4 5 6 7

- 15

- {3,5,7}

- 无要求？选出最大m个。
- 设最大值a[i]不选，a[i-1]选了而a[i+1]不选，此时a[i-1]可换为a[i]。
- 因此若最大值a[i]不选，a[i-1]和a[i+1]一定同时选。
- 选了a[i]，删去a[i-1],a[i],a[i+1]，替换为a[i-1]+a[i+1]-a[i]。
- 意义?
- 撤销对i的选取，并改为i-1和i+1。

- bzoj2151

支持删除的堆？

- 额外增加一个堆存储需要删除的元素。
- 当两个堆堆顶相等时，都弹出。

- 长为n的序列，选择不超过m个连续段，和最大。

- 5 2
- 2 -3 2 -1 2

- 5

- {2},{2,-1,2}

- 线段树维护最大子段和。
- 取完一段，段内*-1，共取m次。

- 合并相邻的同号数，变成正负交替。
- 若正数数量<=m，选所有正数。
- 否则，选择的段数超过m，需要去掉一些。
- 去掉一个正数：总和减去该正数的值，看作该正数不选，段数减一。
- 去掉一个负数：总和减去该负数的绝对值，看作与相邻的正数合并，因此段数也减一。
- 同时，相邻的数不能都去掉。
- 转化为了上一题。

- bzoj2288

- Alice和Bob初始各有一个数字0。
- 长为n的序列 a_i 表示Alice在第 i 天会增加的数字。
- 长为n的序列 b_i 表示Bob在第 i 天会增加的数字。
- 选择一些天进行操作，设当天Alice数字为 x ，Bob数字为 y ，操作会将Alice的数字变为 $\max(0, x-y)$ ，Bob的数字变为 $\max(0, y-x)$ 。
- 哪个人先至少拥有了k分就输。
- 给出一个让A获胜的方案。
- $n \leq 2 \times 10^5$
- 11 17
- 5 2 8 2 4 6 1 2 7 2 5
- 4 6 3 3 5 1 7 4 2 5 3
- 2 4

- 设 a_i 的前缀和为 A_i ， b_i 的前缀和为 B_i 。
- 容易发现进行操作并不会改变两人的差值，只会都减去二者中较小的数，也就是让较小数变为0。
- 因此，要知道第 i 天两人的分数，不需要知道具体在哪些天发生了操作，只需知道之前最近一次发生操作的天数 j ，此时二者的数字为：
- $A_i - \min\left(A_j, B_j\right), B_i - \min(A_j, B_j)$
- 让Alice获胜，即 $A_i - \min\left(A_j, B_j\right) < k, B_i - \min\left(A_j, B_j\right) \geq k$ 。
- 即 $A_i - k < \min\left(A_j, B_j\right) \leq B_i - k$ 。
- 枚举 i ，查找是否有这样的 j 。
- 有了最小可能天数，只需再模拟一次即可知道方案。
- $O(nlogn)$
- cf1250G

- 给定一个长度为N的序列，允许翻转一个子序列，求最长不下降子序列长度。
- N<=50
- 9
- 1 2 3 9 5 6 8 7 4

- 9
- {9,8,7,4}

- 翻转子序列?
- 从两边往中间看，每次交换首尾两数。
- f[l][r][i][j]：区间[l,r]，值域[i,j]

- bzoj4758

- 求两个字符串的最长公共子序列，以及最长公共子序列个数。
- N<=5000

- ABCBDAB
  BACBBD

  4
- 7

- s[i]==t[j],f[i][j]=f[i-1][j-1]+1,g[i][j]+=g[i-1][j-1]
- f[i][j]==f[i-1][j],g[i][j]+=g[i-1][j]
- f[i][j]==f[i][j-1],g[i][j]+=g[i][j-1]
- f[i][j]==f[i-1][j-1],g[i][j]-=g[i-1][j-1]

- bzoj2423

- 给定n，求所有n!种排列，最长上升子序列的长度之和。

- 3
- 12

- {1,2,3}=3
- {1,3,2}=2
- {2,1,3}=2
- {2,3,1}=2
- {3,1,2}=2
- {3,2,1}=1

- 设f[i]表示前i个数，最长上升子序列长度。
- 将f[i]差分后变为01序列，状压。
- 枚举数字i+1插入到哪里。
- 设插入到第j位，该位的差分位1（i+1是当前最大）
- 后续的第一个1变为0（相同）

- bzoj5161

- 判断用两个栈能否将序列排序。

- 若$i<j<k$且$a[k]<a[i]<a[j]$，则i与j不能在同一个栈。
- 二分图染色。

- NOIP2008