基本数据结构 - いち

栈

栈：本质就是一个多层无限高的盒子，每层只能装一个数，只能从最顶部放入/取出数。

一句话来说就是先进先出。

应用主要是括号匹配和后缀表达式求值。

单调栈：保证盒子内部的数是从小到大或从大到小。

干讲不如看题。

P5788 【模板】单调栈

给出项数为 n 的整数数列 $a_{1\dots n}$。

定义函数 $f(i)$ 代表数列中第 i 个元素之后第一个大于 a_i 的元素的**下标**，即 $f(i)=\min_{i<j\le n,a_j>a_i}\{j\}$。若不存在，则 $f(i)=0$。

试求出 $f(1\dots n)$。

对于 100% 的数据，$1\le n\le 3\times 10^6$，$1\le a_i\le 10^9$。

```
5
1 4 2 3 5
```

Sol

维护一个从底到顶递减的单调栈，现在查某一个数的答案，将栈顶小于它的数全部弹出。

此时栈顶就是第一个大于这个数的数。再要查的数加入栈中。

三水的栈

维护一个栈，支持查询栈中最大值和次大值。

Sol

入栈出栈可以视为一个建树的过程，入栈即是对当前节点新开一个子节点，出栈就是回到父节点。

树上前缀最大和次大？随便维护一下啦！

或者也可以将最大值单独维护，其他的开个set。

文本操作

维护一段文本和光标，支持光标处插入删除，查询光标前的字符以及前后移动光标。

Sol

光标前后都是一个栈，开一前一后两个栈就好了。（这玩意叫对顶栈）

P4147 玉蟾宫

这片土地被分成 $N \times M$ 个格子，每个格子里写着 'R' 或者 'F'，R 代表这块土地被赐予了 rainbow，F 代表这块土地被赐予了 freda。

现在 freda 要在这里卖萌。。。它要找一块矩形土地，要求这片土地都标着 'F' 并且面积最大。

但是 rainbow 和 freda 的 OI 水平都弱爆了，找不出这块土地，而蓝兔也想看 freda 卖萌（她显然是不会编程的……），所以它们决定，如果你找到的土地面积为 S，它们每人给你 S 两银子。
对于 100% 的数据，$1 \leq N, M \leq 1000$。

Sol

预处理出每个 'F' 格子向上有多少个 'F' 格子，每行向左向右分别跑一遍单调栈即可。

队列

一队数，在队头加，队尾出。就是先进先出。

单调队列就是队列里是单调的。

P1886 滑动窗口 /【模板】单调队列

有一个长为 n 的序列 a，以及一个大小为 k 的窗口。现在这个从左边开始向右滑动，每次滑动一个单位，求出每次滑动后窗口中的最大值和最小值。

例如，对于序列 $[1,3,-1,-3,5,3,6,7]$ 以及 $k=3$，有如下过程：

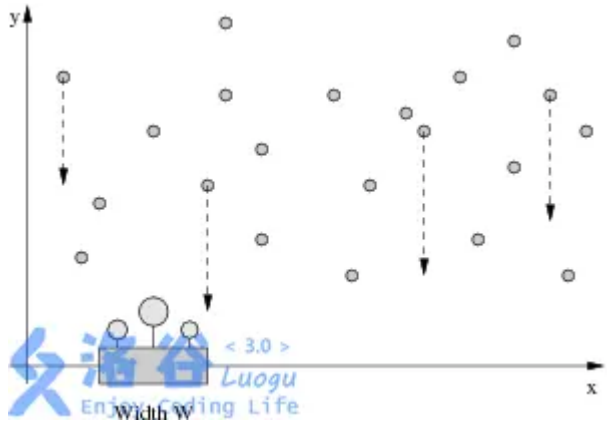
窗口位置	最小值	最大值
[1 3 -1] -3 5 3 6 7	-1	3
1 [3 -1 -3] 5 3 6 7	-3	3
1 3 [-1 -3 5] 3 6 7	-3	5
1 3 -1 [-3 5 3] 6 7	-3	5
1 3 -1 -3 [5 3 6] 7	3	6
1 3 -1 -3 5 [3 6 7]	3	7

对于 100% 的数据，$1 \leq k \leq n \leq 10^6$，$a_i \in [-2^{31}, 2^{31})$。

Sol

求最小值时，维护一个单调队列（队首为最小，队尾为最大）
窗口每次向右移动一次，从队首依次弹出已经不在窗口内的，队尾加入窗口移动后新增的数。

P2698 [USACO12MAR] Flowerpot S



老板需要你帮忙浇花。给出 N 滴水的坐标，y 表示水滴的高度，x 表示它下落到 x 轴的位置。

每滴水以每秒 1 个单位长度的速度下落。你需要把花盆放在 x 轴上的某个位置，使得从被花盆接着的第 1 滴水开始，到被花盆接着的最后 1 滴水结束，之间的时间差至少为 D。

我们认为，只要水滴落到 x 轴上，与花盆的边沿对齐，就认为被接住。给出 N 滴水的坐标和 D 的大小，请算出最小的花盆的宽度 W。

100% 的数据：$1 \leq N \leq 10^5$ ，$1 \leq D \leq 10^6$ ，$0 \leq x, y \leq 10^6$ 。

Sol

先将所有水滴按照 x 排序。

维护一个长度可变的窗口，最开始窗口的 l, r 都在最左边。

然后将窗口的 r 向右移动直到满足条件。

再将窗口的 l 向右移动一个水滴，类似滑动窗口的处理方式，若移动的水滴的高度最小，移走当前水滴后可能不满足条件，将右边界向右移动直到满足条件。

复杂度？l，r都各自只会向右移动 n 次，所以是 $O(n)$ 的。

[POI2015] WIL

给定一个长度为 n 的序列，你有一次机会选中一段连续的长度不超过 d 的区间，将里面所有数字全部修改为 0。请找到最长的一段连续区间，使得该区间内所有数字之和不超过 p。

Sol

首先可知随区间右端点增大，左端点一定单调不减。
前缀和优化一下区间和，再用单调队列维护区间内长度为 d 的值最大的区间判断是否可行，不可行则将左端点右移。

- STL中的优先队列，实际上是建了一个大根堆

priority_queueq;

插入操作：q.push(a);

弹出堆顶操作：q.pop();

堆顶元素：q.top();

如果要建一个小根堆，可以将数全部添个负号在丢进堆里。

也可以用结构体重载运算符。

上面两种数据结构很少单独设题，常见于DP的优化。

P1725 琪露诺

在幻想乡，琪露诺是以笨蛋闻名的冰之妖精。

某一天，琪露诺又在玩速冻青蛙，就是用冰把青蛙瞬间冻起来。但是这只青蛙比以往的要聪明许多，在琪露诺来之前就已经跑到了河的对岸。于是琪露诺决定到河岸去追青蛙。

小河可以看作一列格子依次编号为 0 到 N，琪露诺只能从编号小的格子移动到编号大的格子。而且琪露诺按照一种特殊的方式进行移动，当她在格子 i 时，她只移动到区间 $[i+L,i+R]$ 中的任意一格。你问为什么她这么移动，这还不简单，因为她是笨蛋啊。

每一个格子都有一个冰冻指数 A_i，编号为 0 的格子冰冻指数为 0。当琪露诺停留在那一格时就可以得到那一格的冰冻指数 A_i。琪露诺希望能够在到达对岸时，获取最大的冰冻指数，这样她才能狠狠地教训那只青蛙。

但是由于她实在是太笨了，所以她决定拜托你帮它决定怎样前进。

开始时，琪露诺在编号 0 的格子上，只要她下一步的位置编号大于 N 就算到达对岸。

对于 100% 的数据，$N\le 2\times 10^5$，$-10^3\le A_i\le 10^3$，$1\le L\le R\le N$。数据保证最终答案不超过 $2^{31}-1$。

Sol

令 f_i 为走到编号 i 的格子得到的最大冰冻指数。

$f_i=\max_{j=i-L}^{i-R}(f_j)+A_i$

每一次能够转移到 f_i 的是一个窗口，维护一下窗口中的最大值即可。

单调队列优化dp的习题：

P2627 [USACO11OPEN] Mowing the Lawn G

P3572 [POI2014] PTA-Little Bird

思考题：怎么用两个栈维护队列的功能？

队列支持队头加入队尾弹出，而栈仅支持栈顶加入弹出。

维护两个栈，一个加入栈，一个删除栈。

当删除栈空了的时候将加入栈依次弹入删除栈。

堆

堆是一棵树，其每个节点都有一个键值，且每个节点的键值都大于等于/小于等于其父亲的键值。

大于等于就叫大根堆，小于等于就叫小根堆。

一般的操作有：插入一个值、查询最值，删除最值，合并堆。

一些功能强大的堆（可并堆/左偏树）还可以高效地合并。

有一些堆也可以可持久化。

STL中优先队列的实现就是一个二叉堆。

二叉堆具体结构：

满足堆的性质，且是一颗完全二叉树。

以小根堆为例。

插入操作：在最下一层最右边的节点插入，如果最下一层已满，新增一层。新加节点的权值小于其父亲的权值，就不断向上与父亲交换。

删除操作：就是删根节点，然后将最后一个节点拿到根节点上，在该节点的儿子上，找一个最小的与该节点交换，不断重复操作。

可以发现操作前后都满足完全二叉树的性质，所以一次操作复杂度是 $O(logn)$ 的。

Que いち

维护一个序列，支持以下操作：

插入一个元素，询问第k大，删除第k大，将k加一或减一。

Sol

维护两个堆，一个堆维护前 $k-1$ 大，一个堆维护其他的数。

弱化版的题：P1801 黑匣子

左偏树/可并堆

考虑一般的数据结构怎么合并?

有一种叫启发式合并的方法，就是合并两个结构时，将规模较小的结构拆成单点依次插入规模较大的结构中。

对应到堆上，就是若有一大一小两个堆合并，将小堆中的元素依次取出来插入大堆。

两个堆合并时，合并一次的复杂度是小堆的大小乘个log，小堆的大小小于合并成的堆的一半，因此极限情况是类似线段树节点的合并。总复杂度是两只log。

对大多数数据结构都可以这样合并，不过会在原有的时间复杂度上套个log。

那我们想要单log的堆合并?

P3377 【模板】左偏树/可并堆

如题，一开始有 n 个小根堆，每个堆包含且仅包含一个数。接下来需要支持两种操作：

1. `1 x y`：将第 x 个数和第 y 个数所在的小根堆合并（若第 x 或第 y 个数已经被删除或第 x 和第 y 个数在同一个堆内，则无视此操作）。
2. `2 x`：输出第 x 个数所在的堆最小数，并将这个最小数删除（若有多个最小数，优先删除先输入的；若第 x 个数已经被删除，则输出 -1 并无视删除操作）。

对于 100% 的数据：$n \leq 10^5$，$m \leq 10^5$，初始时小根堆中的所有数都在 `int` 范围内。

Sol

定义一个节点的 dis_u 是节点 u 到子树内最近的叶子节点的距离。

左偏树不仅满足堆的性质，是一棵二叉树，也满足所有节点左儿子的 dis 大于右儿子的。

若要合并两树 x 和 y ,令 x 为堆顶更小的树，要将 y 合并到 x 上。将 x 的根作为合并后树的根，然后将这个根的左儿子作为合并后堆的左儿子，递归地合并其右儿子与另一个堆，作为合并后的堆的右儿子。为了满足左偏性质，合并后若左儿子的 dis 小于右儿子的，就交换两个儿子。

由于左偏性质，每递归一层其中一个堆的根节点的 dis 就会减少1，根的 dis 不超过 $logn$，所以合并的复杂度是单log的。

插入节点：单节点作为一个堆合并。

删除堆顶：删掉堆顶后合并左右儿子。

甚至可以做整个堆加减和乘一个值，只要打的标记改变堆中元素的相对大小就行。

据说还有什么随机交换左右儿子的算法，平均时间复杂度为log的（~~但是我不会证~~），谨慎使用。

多倍经验：

P2713 罗马游戏

P1456 猴王出世

还有配对堆和斐波那契堆可以 $O(1)$ 合并两个堆。有兴趣自行了解。

P1552 [APIO2012] 派遣

在这个帮派里，有一名忍者被称之为 Master。除了 Master 以外，每名忍者都有且仅有一个上级。为保密，同时增强忍者们的领导力，所有与他们工作相关的指令总是由上级发送给他的直接下属，而不允许通过其他的方式发送。

现在你要招募一批忍者，并把它们派遣给顾客。你需要为每个被派遣的忍者支付一定的薪水，同时使得支付的薪水总额不超过你的预算。另外，为了发送指令，你需要选择一名忍者作为管理者，要求这个管理者可以向所有被派遣的忍者发送指令，在发送指令时，任何忍者（不管是否被派遣）都可以作为消息的传递人。管理者自己可以被派遣，也可以不被派遣。当然，如果管理者没有被排遣，你就不需要支付管理者的薪水。

你的目标是在预算内使顾客的满意度最大。这里定义顾客的满意度为派遣的忍者总数乘以管理者的领导力水平，其中每个忍者的领导力水平也是一定的。

写一个程序，给定每一个忍者 i 的上级 B_i，薪水 C_i，领导力 L_i，以及支付给忍者们的薪水总预算 M，输出在预算内满足上述要求时顾客满意度的最大值。

$1 \leq N \leq 10^5$，$1 \leq M \leq 10^9$，$0 \leq B_i < i$，$1 \leq C_i \leq M$，$1 \leq L_i \leq 10^9$。

简化题面：n个点组成一棵树，每个点都有一个领导力和费用，可以让一个点当领导，然后在这个点的子树中选择一些费用之和不超过m的点，得到领导的领导力乘选择的点的个数（领导可不被选择）的利润。求利润最大值。

Sol

从叶子节点开始，每个点开始是一个大根堆，堆里的就是这个点的人。

往父亲那里合并堆，记录堆的大小，费用的总和。

从儿子合并完毕后，在每个节点，不断踢出费用最大的人，直到费用的总和<=m 这就是这个点的最优方案了。（显然，花费最小的都留下了）

对于每个点，用这个点的领导力乘堆的大小尝试更新答案即可。

并查集

que：有多个集合，要求快速找到两个元素是否在同一个集合内和合并集合。

为每个元素设置一个 fa_u，代表 u 和 fa_u 同属于一个集合。初始时 $fa_u = u$。

显然某一时刻给 u 和 fa_u 连边，可以得到一个森林。同属一棵树的在一个集合内。

合并两个集合时，只需要将其中一个集合的对应根节点的 fa 设置为另一个集合的根节点。

若要找到两个元素是否在同一集合，只需判断对应树的根节点是否相同。（不断跳fa[u]找根节点）

有特殊情况能卡掉上述算法，如构造树为一条链，不断查询链底元素。

因而有两个小优化：
1.在跳fa_u时，将路径上的所有点的fa_u都设为根节点。
2。启发式合并：将对应树大小较小的树根节点fa_u设置为较大的树的根节点。

核心代码：

```
int getfa(intx){
    if(fa[x]==x)return x;
    return fa[x]=getfa(fa[x]);
}
```

启发式合并顺便维护一个树大小就行，树的大小保存在根节点上。

P1525 [NOIP2010 提高组] 关押罪犯

S 城现有两座监狱，一共关押着 N 名罪犯，编号分别为 $1 \sim N$。他们之间的关系自然也极不和谐。很多罪犯之间甚至积怨已久，如果客观条件具备则随时可能爆发冲突。我们用“怨气值”（一个正整数值）来表示某两名罪犯之间的仇恨程度，怨气值越大，则这两名罪犯之间的积怨越多。如果两名怨气值为 c 的罪犯被关押在同一监狱，他们俩之间会发生摩擦，并造成影响力为 c 的冲突事件。

每年年末，警察局会将本年内监狱中的所有冲突事件按影响力从大到小排成一个列表，然后上报到 S 城 Z 市长那里。公务繁忙的 Z 市长只会去看列表中的第一个事件的影响力，如果影响很坏，他就会考虑撤换警察局长。

在详细考察了 N 名罪犯间的矛盾关系后，警察局长觉得压力巨大。他准备将罪犯们在两座监狱内重新分配，以求产生的冲突事件影响力都较小，从而保住自己的乌纱帽。假设只要处于同一监狱内的某两个罪犯间有仇恨，那么他们一定会在每年的某个时候发生摩擦。

那么，应如何分配罪犯，才能使 Z 市长看到的那个冲突事件的影响力最小？这个最小值是多少？

对于 100% 的数据有 $N \leq 20000, M \leq 100000$。

Sol

敌人的敌人就是朋友，遵照这个规则按影响力从大到小排序依次加入并查集。

当两个冲突的罪犯不得不在同一个并查集中，此时的影响力就是答案。

并查集最常见的一个应用就是求最小生成树。

下面是道最小生成树的题。

P1967 [NOIP2013 提高组] 货车运输

A 国有 n 座城市，编号从 1 到 n，城市之间有 m 条双向道路。每一条道路对车辆都有重量限制，简称限重。

现在有 q 辆货车在运输货物， 司机们想知道每辆车在不超过车辆限重的情况下，最多能运多重的货物。

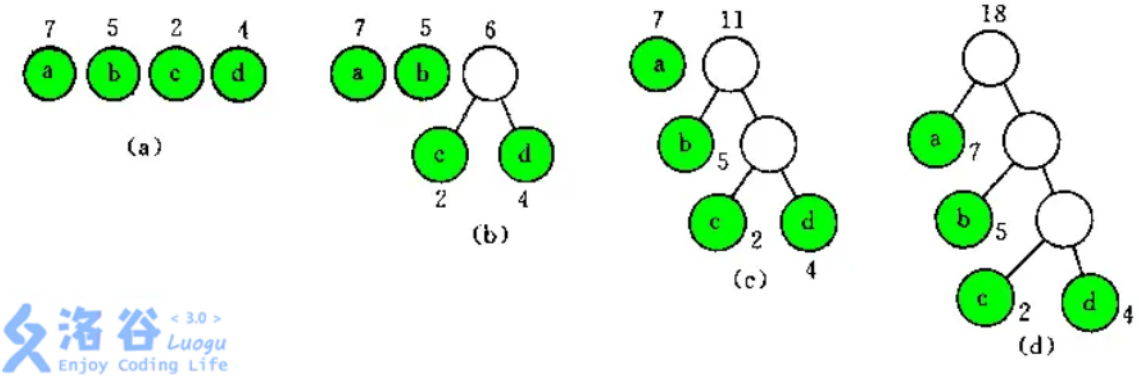
对于 100% 的数据，$1 \leq n < 10^4$，$1 \leq m < 5 \times 10^4$，$1 \leq q < 3 \times 10^4$，$0 \leq z \leq 10^5$。

对于最小生成树的拓展kruskal重构树，可以看看这道题

哈夫曼树

现在有n个点，每个点有权值，路径长度为1，构造二叉树令这n个点都在叶子上，且使得 单点权值 × 这个点到根节点的路径长 的和最小。

算法本质是个贪心，每次选根节点权值最小的两棵树合并，将合并后根的权值设为原两树权值和，重复上述过程直到只剩一棵树



拓展：k叉哈夫曼树

就是每次选权值前k小的节点合并。

这玩意就在noi2015考过一次，大概率是论文题。。。了解一下就行