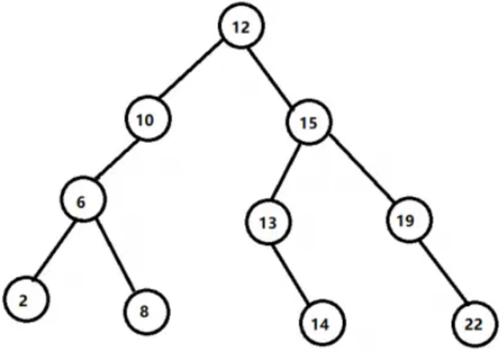
# 问题

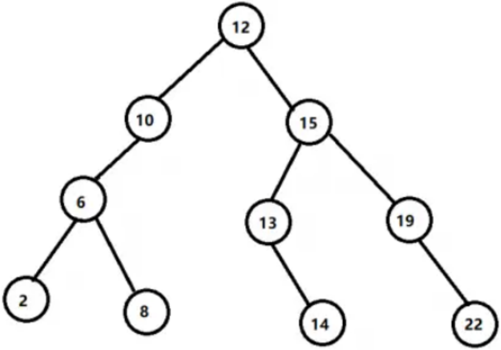
- 动态地维护一个可重集合 $M$，支持以下操作：
  - 向 $M$ 中插入一个数 $x$。
  - 从 $M$ 中删除一个数 $x$（若有多个相同的数，只删除一个）。
  - 查询 $M$ 中有多少个数比 $x$ 小。
  - 查询 $M$ 中排名为 $x$ 的数。
  - 查询 $M$ 中 $x$ 的前驱。
  - 查询 $M$ 中 $x$ 的后继。

# 二叉搜索树



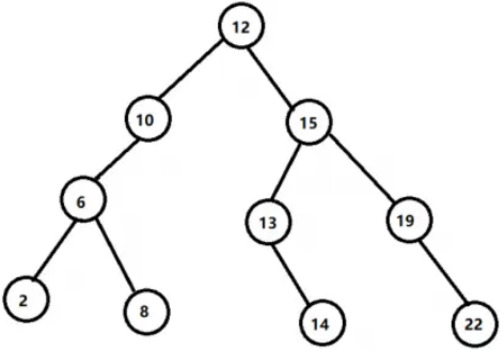

- 对于每个结点 $x$，其左子树所有点的权值小于 $x$ 的权值，右子树所有点的权值大于 $x$ 的权值。重复元素可以记录出现次数，以下操作假定不存在重复元素。

# 二叉搜索树



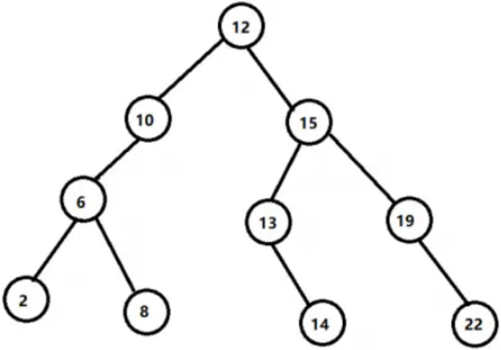

- 插入：根据与当前点的大小关系向左右子树递归，作为一个新的叶子结点。
- 删除：找到对应点，如有两棵子树，则用左子树最大值代替当前点。

# 二叉搜索树




- 求排名：找到 $v$ 对应的点，在路径上若进入右子树，说明左子树所有点权值都更小，加上这些点的数量 $sz_{ls}+1$ 即可。
- 求排名为 $k$ 的点：逆过程，如果左子树大小 $\leq k$ 则向左递归，否则在右子树中查找排名为 $k-sz_{ls}-1$ 的点。

# 二叉搜索树




- 求前驱：找到 $v$ 对应的点，在其左子树内一直向右递归走到的点即为前驱。
- 求后继：找到 $v$ 对应的点，在其右子树内一直向左递归走到的点即为后继。

# 二叉搜索树

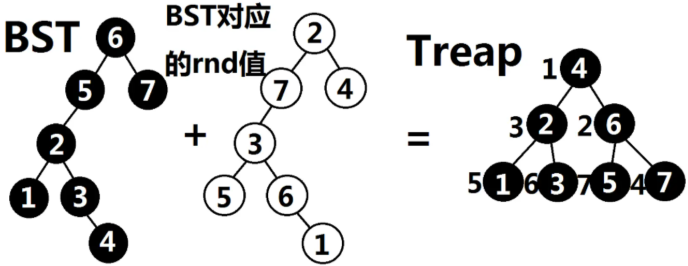
- 时间复杂度：一次操作最坏情况是 $O(n)$ 的，例如二叉搜索树形成一条链的情况。
- 但是对于同样的元素集合，可以构建出很多不同的二叉搜索树，我们希望让这棵树尽可能地“平衡”。由此引出了平衡树，通过一定操作维持树的高度（平衡性）来降低操作的复杂度。
- 接下来大致介绍一些平衡树的具体思路。

# 替罪羊树

- 给定一些数，我们可以通过取中点分治构造出高度为 $O(\log n)$ 的二叉搜索树，但是插入删除可能破坏平衡性。替罪羊树是暴力重构的思想，每次操作过后，找到操作路径上最浅的满足 $\max_{sz_{ls}, sz_{rs}} > \alpha \cdot sz_x$ 的点 $x$，然后重构 $x$ 的子树。$\alpha$ 一般取 $0.7 \sim 0.8$，此时复杂度可以看做均摊 $O(n \log n)$。

# Treap

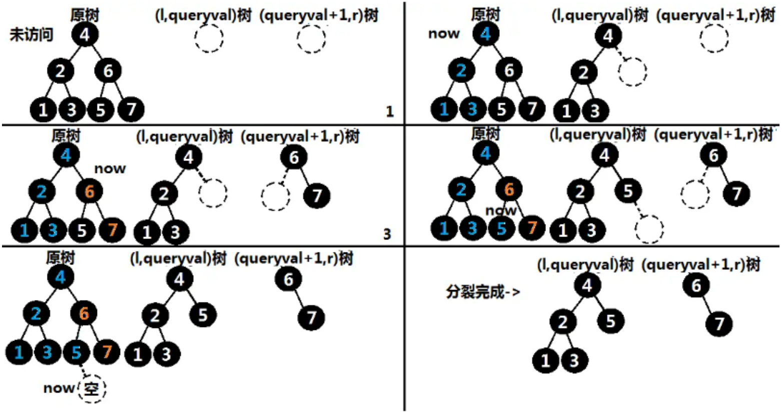
- 仍然考虑分治建树的过程，考虑给每个元素带上随机的权值 $rnd_x$，每次分治时取当前分值区间随机权值最小的点作为根节点，递归两侧建树。这样建出的树满足 $x$ 子树内所有点的 $rnd$ 值都比 $rnd_x$ 更大。可以证明这样建出的树期望高度是 $O(\log n)$ 的。
- Treap 即在插入元素的时候给其带上上文中的随机权值，时刻维护一棵满足上述性质的树。



# FHQ Treap

- 主要讲讲 FHQ Treap，通过分裂和合并操作维护一棵 Treap。
- 分裂：给定一个权值 $v$，将原树分成两棵树，一棵树所有权值 $\leq v$，另一棵树所有权值 $> v$，同时这两棵树也满足 Treap 的限制关系。
- 合并：给定两棵 Treap，其中一棵内的所有点权值 max 小于另一棵内所有点权值 min，将这两棵树合并成满足限制的一棵 Treap。

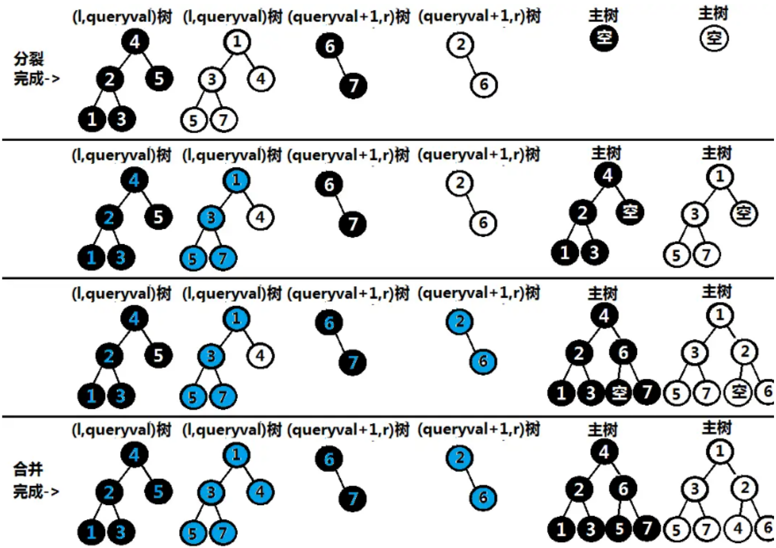
# FHQ Treap



# FHQ Treap

```
void split(int now,int k,int &x,int &y) {
    if(!now) {
        x=y=0;
        return ;
    }
    if(a[now].val<=k) {
        x=now;
        split(a[now].rs,k,a[x].rs,y);
    } else {
        y=now;
        split(a[now].ls,k,x,a[y].ls);
    }
    pushup(now);
}
```

# FHQ Treap



# FHQ Treap

```
int merge(int x,int y) {
    if(!x||!y) {
        return x+y;
    }
    if(a[x].rnd<a[y].rnd) {
        a[x].rs=merge(a[x].rs,y);
        pushup(x);
        return x;
    } else {
        a[y].ls=merge(x,a[y].ls);
        pushup(y);
        return y;
    }
}
```

# FHQ Treap

- 插入 $v$：按 $v$ 分裂，将 $v$ 当做只有一个点的树，再合并三棵树。
- 删除 $v$：按 $v$ 分裂，再按 $v-1$ 分裂，合并 $v$ 的左右儿子（只删除一个 $=v$ 的点），再合并三棵树。
- 剩下的操作都可以在当前 Treap 上暴力去做，也可以复用分裂合并简化代码。

# 按子树大小分裂

- 文艺平衡树：多次翻转序列中 $[l, r]$ 部分（reverse）。

# 按子树大小分裂

- 文艺平衡树：多次翻转序列中 $[l, r]$ 部分（reverse）。
- 按照子树大小进行分裂，分裂出前 $k$ 个元素：

```
void split(int now,int k,int &x,int &y) {
    if(!now) {
        x=y=0;
        return ;
    }
    pushdown(now);
    if(k<=a[a[now].ls].size) {
        y=now;
        split(a[now].ls,k,x,a[y].ls);
    } else {
        x=now;
        split(a[now].rs,k-a[a[now].ls].size-1,a[x].rs,y);
    }
    pushup(now);
}
```

# 按子树大小分裂

- 翻转：交换左右儿子，打懒标记即可。
- 整棵树的中序遍历就是最终的序列。
- 在线段树的基础上，可以额外维护单点插入、区间翻转等操作。

# 题目

**CF702F T-Shirts**

有 $n$ 种衣服，每种有价格 $c_i$ 和品质 $q_i$。$m$ 个人要买衣服，第 $i$ 个人有 $v_i$ 元，每人每次都会买一件能买得起的 $q_i$ 最大的衣服。一个人只能买一种衣服一件，所有人之间都是独立的。问最后每个人买了多少件衣服？如果有多个 $q_i$ 最大的衣服，会从价格低的开始买。

$n, m \le 2 \times 10^5$。

# 题目

## CF702F T-Shirts

有 $n$ 种衣服，每种有价格 $c_i$ 和品质 $q_i$。$m$ 个人要买衣服，第 $i$ 个人有 $v_i$ 元，每人每次都会买一件能买得起的 $q_i$ 最大的衣服。一个人只能买一种衣服一件，所有人之间都是独立的。问最后每个人买了多少件衣服？如果有多个 $q_i$ 最大的衣服，会从价格低的开始买。
$n, m \leq 2 \times 10^5$。

- 考虑将衣服排序，同时维护每个人的剩余钱数。假设当前价格为 $c$，每次将剩余钱数按照 $[1, c), [c, 2c), [2c, \infty)$ 分段处理。
- 暴力处理 $[c, 2c)$，每次钱数至少减半，总复杂度 $O(n \log n \log v)$。

# 题目选讲

**CF220B Little Elephant and Array**

给定长为 $n$ 的序列 $a$，$m$ 次询问给出 $l, r$，求 $[l, r]$ 内有多少个数 $a_i$ 在 $[l, r]$ 内的出现次数恰为 $a_i$。$n, m \leq 10^5$。

# 题目选讲

**CF220B Little Elephant and Array**

给定长为 $n$ 的序列 $a$，$m$ 次询问给出 $l,r$，求 $[l,r]$ 内有多少个数 $a_i$ 在 $[l,r]$ 内的出现次数恰为 $a_i$。$n,m \le 10^5$。

- 一个数 $x$ 可能对一个询问产生贡献的必要条件的是其出现过至少 $x$ 次，这样的数不超过 $\sqrt{n}$ 个，前缀和维护即可。

# 题目选讲

**P3793 由乃救爷爷**

给定长为 $n$ 的序列 $a$，$m$ 次询问区间最大值。序列和询问都随机生成。

$n, m \leq 2 \times 10^7$。

# 题目选讲

**P3793 由乃救爷爷**

给定长为 $n$ 的序列 $a$，$m$ 次询问区间最大值。序列和询问都随机生成。
$n, m \leq 2 \times 10^7$。

- 分块，处理块间 ST 表和块内前后缀最值。随机询问落到同一块内的概率是 $\frac{B}{n}$，此时询问复杂度为 $O(B)$，故可取 $B = O(\sqrt{n})$，复杂度期望 $O(n)$。

# 题目选讲

**CF1000F One Occurrence**

给定长为 $n$ 的序列 $a$，$q$ 次询问，每次给定一个区间，找出其中只出现了一次的数。
$n, q \leq 5 \times 10^5$。

# 题目选讲

**CF1000F One Occurrence**

给定长为 $n$ 的序列 $a$，$q$ 次询问，每次给定一个区间，找出其中只出现了一次的数。
$n, q \leq 5 \times 10^5$。

- 按右端点从小到大扫描，对于每种数只保留最后一次出现的位置，那么 $x$ 只出现一次，等价于 $x$ 在区间内且 $x$ 前一次出现在区间外，线段树维护保留下来的位置中每个数前一次的出现位置即可。

# 题目选讲

## P10833 [COTS 2023] 下 Niz

给定长为 $n$ 的序列 $a$，求有多少个区间 $[l, r]$，满足区间内的元素恰为 $1 \sim r - l + 1$ 的一个排列。

$n \leq 10^6$。

# 题目选讲

**P10833 [COTS 2023] 下 Niz**

给定长为 $n$ 的序列 $a$，求有多少个区间 $[l, r]$，满足区间内的元素恰为 $1 \sim r - l + 1$ 的一个排列。
$n \leq 10^6$。

- 区间最大值确定了，区间长度就确定了。在最大值处进行分治，每次枚举两侧较短的一边，区间就能直接确定下来，再判定即可。这里枚举的复杂度是 $O(n \log n)$ 的，类似启发式合并。
- 判定区间是否合法：只要区间内没有重复出现的数即可，维护每个数上一次出现的位置 $p_i$，若 $\max_{i=l}^{r} p_i \geq l$，则区间 $[l, r]$ 不合法。

# 应用

**UOJ637 【美团杯 2021】A. 数据结构**

给定一个长为 $n$ 的序列，$q$ 次询问给 $[l,r]$ 内的数加一之后全局不同的数的数量。

$n,q \le 10^6$。

# 应用

**UOJ637【美团杯 2021】A. 数据结构**

给定一个长为 $n$ 的序列，$q$ 次询问给 $[l,r]$ 内的数加一之后全局不同的数的数量。
$n,q \le 10^6$。

- 考察某种元素没有贡献的条件。
- 将偏序关系放到二维平面上考虑，询问 $(l,r)$ 作为平面上的一个点。对于元素 $x$，将其没有贡献的条件表示为一个平面上的矩形：$x-1$ 不在区间内是若干矩形的并（$pos_i < l, r < pos_{i+1}$），$x$ 不在区间外是在切割上述的矩形（$l \le p_x, q_x \le r$，其中 $p_x, q_x$ 表示 $x$ 最靠前和最靠后的出现位置）。

# 题目选讲

## 例题

二维平面上，先给若干条斜率为 $-1$ 的线段对应位置加上某个数，再多次查询给定矩阵内的权值和。
$n, m \leq 10^6$。

# 题目选讲

**例题**

二维平面上，先给若干条斜率为 $-1$ 的线段对应位置加上某个数，再多次查询给定矩阵内的权值和。
$n, m \leq 10^6$。

- 矩形查询先差分掉，按照 $x+y$ 从小到大的顺序进行扫描线。对于一条线段，其超出矩形的部分，要么是 $x$ 轴超出范围，要么是 $y$ 轴超出范围，不会同时超出。这样的话我们减去超出部分的权值即可，是一维的偏序问题。
- 启示：扫描线可以按照不同的方向处理。

# 题目选讲

**P4141 消失之物**

给定 $n$ 个物品，每个物品有体积。对于所有 $i,j$，求出去掉第 $i$ 个物品后，有多少种方案装满大小为 $j$ 的背包。
$n,m \le 2\times 10^3$。

# 题目选讲

## P4141 消失之物

给定 $n$ 个物品，每个物品有体积。对于所有 $i,j$，求出去掉第 $i$ 个物品后，有多少种方案装满大小为 $j$ 的背包。
$n,m \le 2\times 10^3$。

- 在进行背包 DP 的过程中，放入物品的顺序是不重要的，因此对于要求答案的 $i$，我们可以视为第 $i$ 个物品是最后放入的。记放入第 $i$ 个物品前时 DP 数组为 $g$，放入所有物品时 DP 数组为 $f$。我们希望通过 $f$ 倒推出 $g$。
- 正向转移时 $f_x=\begin{cases} g_x+g_{x-w} & x\ge w \\ g_x & x<w \end{cases}$，因此对于 $x<w$ 可以直接确定 $g_x$，剩余的通过 $g_x=f_x-g_{x-w}$ 得出，从小到大枚举 $x$ 求解即可。

# 题目选讲

**CF1442D Sum**

给定 $n$ 个不降序列，每次操作可以选择任一序列开头的元素删去。进行恰好 $k$ 次操作，最大化删去的数的和。

$n,k \le 3\times 10^3$，$k \le \sum_{i=1}^{n} len_i \le 10^6$。

# 题目选讲

**CF1442D Sum**

给定 $n$ 个不降序列，每次操作可以选择任一序列开头的元素删去。进行恰好 $k$ 次操作，最大化删去的数的和。
$n,k \le 3\times 10^3$，$k \le \sum_{i=1}^{n} len_i \le 10^6$。

- 结论：最多只有一个序列删去了部分数，其他要么不删要么全删。考虑调整法证明。
- 枚举删了部分的序列，剩下的序列是一个背包问题。但此题中是求最值问题，没法用回退背包。考虑分治，假设当前分治区间是 $[l,r]$，我们维护 $[1,l)\cup(r,n]$ 的背包，可以递归处理。

# 题目选讲

## P6189 [NOI Online #1 入门组] 跑步

给定 $n$，求有多少个不降序列 $a$ 满足 $\sum a_i = n$，对给定模数取模。
$n \le 10^5$。

# 题目选讲

**P6189 [NOI Online #1 入门组] 跑步**

给定 $n$，求有多少个不降序列 $a$ 满足 $\sum a_i = n$，对给定模数取模。$n \le 10^5$。

- 直接背包可以做到 $O(n^2)$ 的复杂度。考虑另一种意义的 DP：每次加入一个 1，或是给已有的数全体加 1。对应的转移是 $f_{i,j} = f_{i-1,j-1} + f_{i-j,j}$。
- 对于 $i < \sqrt{n}$ 的数，我们跑完全背包。对于 $i > m$ 的数，我们跑后一种 DP（修改为每次加入一个 $m$）。这样能将时间复杂度平衡至 $O(n\sqrt{n})$。
- 将每个数的大小对应到二维平面上，上面两种 DP 分别对应从横向和纵向切入。

# 题目选讲

## CF1107E Vasya and Binary String

给定一个 01 串，消除一段长度为 $i$ 的全 0 或全 1 串获得分值 $a_i$，消除完后剩下的串会接到一起，求将这个串消除完能获得的最大分数。

$n \le 100$。

# 题目选讲

## CF1107E Vasya and Binary String

给定一个 01 串，消除一段长度为 $i$ 的全 0 或全 1 串获得分值 $a_i$，消除完后剩下的串会接到一起，求将这个串消除完能获得的最大分数。
$n \leq 100$。

- 考虑最后一次进行的操作，会将串分成若干个部分每个部分再独立进行……考虑一步一步地处理这若干个部分，假设我们枚举了第一个部分 $[l, r]$，$[1, l-1]$ 都是跟最后一步操作一起进行的，那么 $[r+1, n]$ 是一个左边带了额外 $l-1$ 个相同数的子问题。
- 设 $f_{l,r,k}$ 表示当前处理区间 $[l, r]$，$l$ 及其左边有连续 $k$ 个相等的数。转移：
  - $f_{l,r,k} \leftarrow f_{l+1,r,1} + a_k$。
  - $f_{l,r,k} \leftarrow [s_l = s_x] f_{l+1,x-1,1} + f_{x,r,k+1}$。

# 题目选讲

**P10982 Connected Graph**

求 $n$ 个点的有标号无向连通图的数量，对 998244353 取模。

$n \le 10^5$

# 题目选讲

**P10982 Connected Graph**

求 $n$ 个点的有标号无向连通图的数量，对 998244353 取模。
$n \le 10^5$

- 枚举 1 号点所在连通块大小，减去不连通图的数量。

$$f_n = 2^{\binom{n}{2}} - \sum_{i=1}^{n-1} \binom{n-1}{i-1} f_i 2^{\binom{n-i}{2}}$$