牛客竞赛
AC.NOWCODER.COM

AC.NOWCODER

2021牛客暑期多校训练营
第 9 场

出题人：华东师范大学

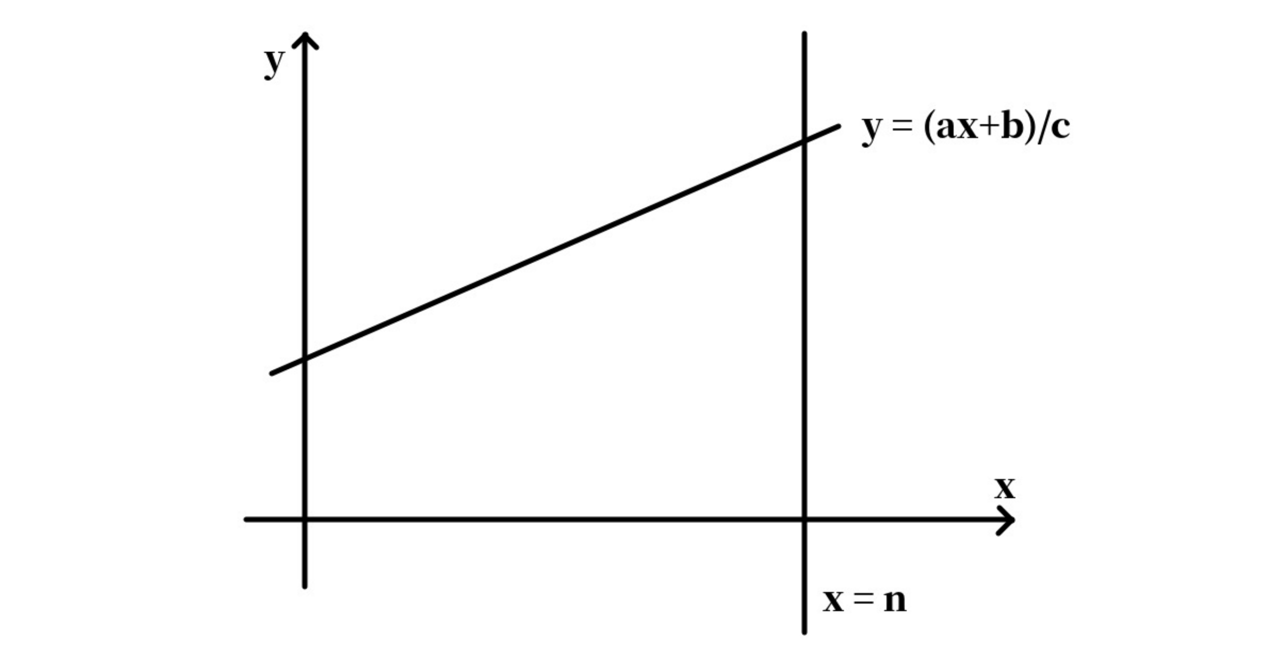
A Math Challenge

牛客竞赛
AC.NOWCODER.COM

- 看起来就很类欧。
- 我们定义一种针对二元函数的变换： $T: P(i,j) \to Q(i,j)$ ，目的是更清晰、严谨地描述算法。
- 对于一个点权和的算式 $\mathrm{F} = \sum_i \sum_j P(i,j)$ ，我们将 T 变换作用于之，得到的是 $T(\mathrm{F}) = \sum_i \sum_j Q(i,j)$。
- 设 $f_{a,b,c,n}(p,q)$ 表示我们要求的答案。
- 那么考虑类欧递归的过程。

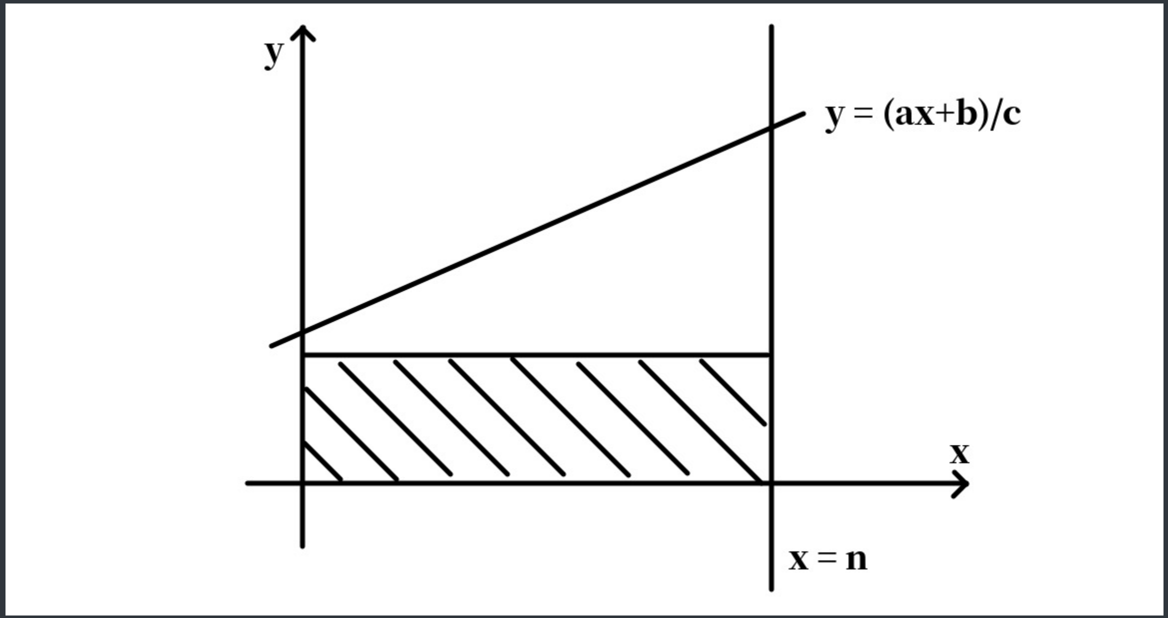
A Math Challenge

- 那么考虑类欧递归的过程。
- 注意到类欧的几何意义是二维平面上一个梯形里的整点点权和。



A Math Challenge (Case 1)

- 如果 $b \geq c$，那么设 $k = \left\lfloor \frac{b}{c} \right\rfloor$。
- 考虑截一个矩形出来，那么矩形的点权和是 $S_r = \sum_{i=0}^{n} \sum_{j=1}^{k} i^p j^q$。



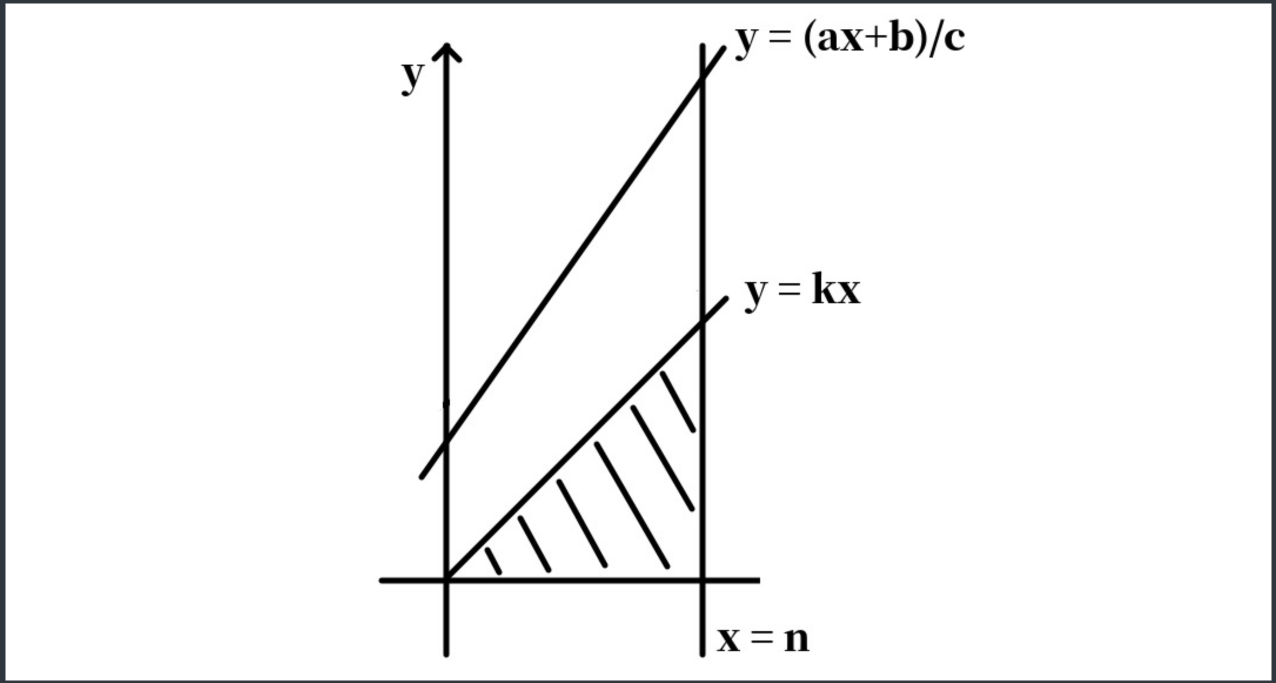
A Math Challenge (Case 1)

牛客竞赛
AC.NOWCODER.COM

- 然后计算矩形上方的梯形。仿照类欧，我们会先计算 $f_{a,(b \bmod c),c,n}$，但实际上这计算的是把上面的梯形平移下来与 x 轴对齐后的点权和。因此我们要对其应用一个变换：$T_1: i^p j^q \to i^p (j+k)^q$。
- 因此 $f_{a,b,c,n}(p,q) = \sum_{i=0}^{n} \sum_{j=1}^{k} i^p j^q + T_1(f_{a,(b \bmod c),c,n})(p,q)$。
- 根据二项式定理：$i^p (j+k)^q = \sum_{t=0}^{q} \binom{q}{t} k^{q-t} i^p j^t$。
- 于是 $f_{a,b,c,n}(p,q) = S_r + \sum_{t=0}^{q} \binom{q}{t} k^{q-t} f_{a,(b \bmod c),c,n}(p,t)$。

A Math Challenge (Case 2)

- 如果 $a \geq c$，那么设 $k = \left\lfloor \frac{a}{c} \right\rfloor$。
- 相当于我们先截一个直角三角形出来，这部分的点权和是 $S_t = \sum_{i=0}^{n} \sum_{j=1}^{ki} i^p j^q$。

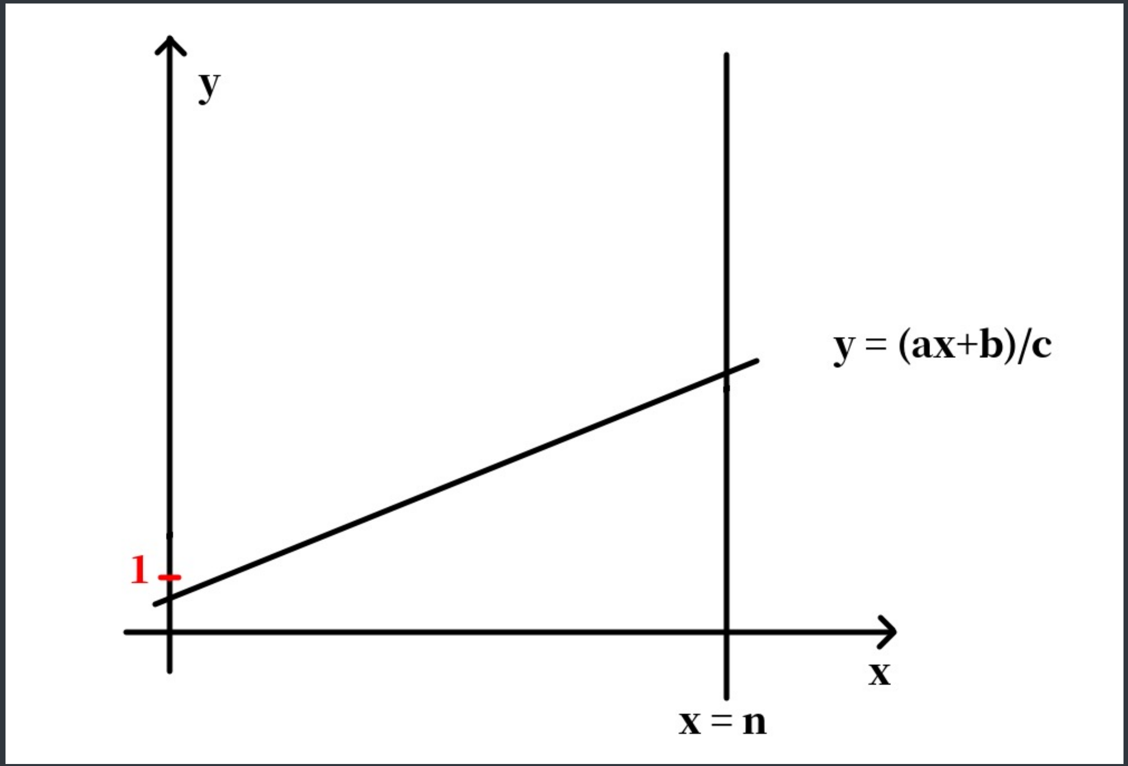


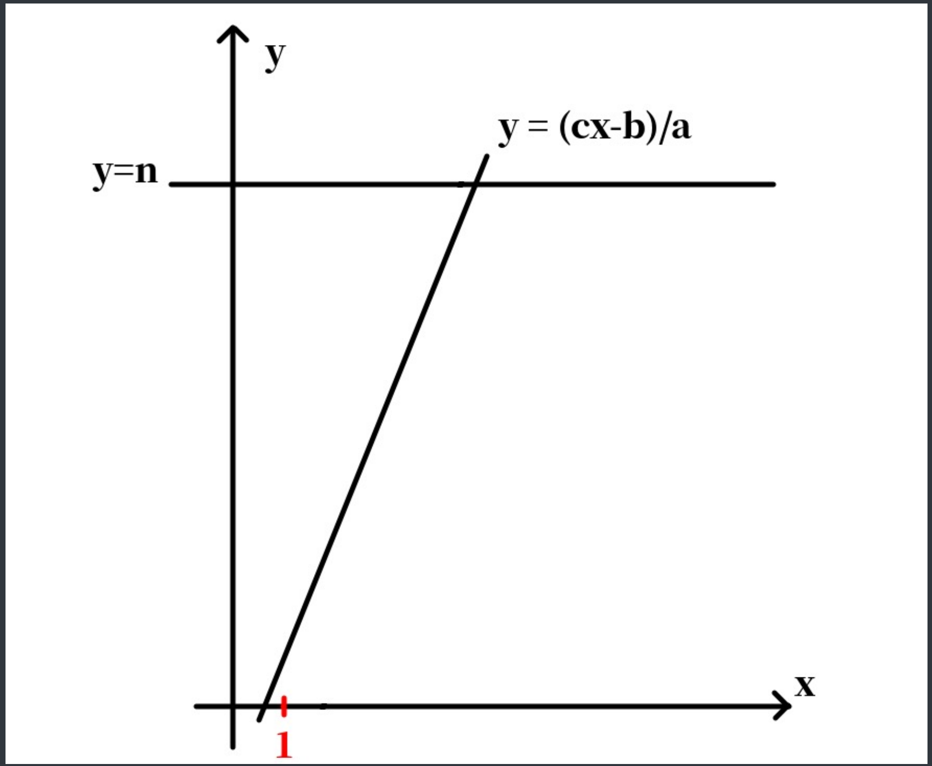
A Math Challenge (Case 2)

- 然后我们对平面上的点做一个变换 $(i,j) \to (i, j-ki)$。这样的话三角形上方的梯形就变成了紧贴 x 轴的直角梯形，这个直角梯形的点权和是 $f_{(a \bmod c),b,c,n}$。
- 不过我们求出了 $f_{(a \bmod c),b,c,n}$ 后，还得把点变回原来的位置，对应的点权变换是 $T_2: i^p j^q \to i^p (j+ki)^q$。
- T_2同样可以用二项式定理展开。
- 于是 $f_{a,b,c,n}(p,q) = S_t + \sum_{t=0}^{q} \binom{q}{t} k^{q-t} f_{(a \bmod c),b,c,n}(p+q-t,t)$。

A Math Challenge (Case 3)

- 如果 $a < c$ 且 $b < c$，那么按照套路我们得将平面沿 $y = x$ 翻转。
- 翻转之后问题转化为矩形的点权和减去三角形点权和。





A Math Challenge (Case 3)

- 这时我们再将整个图形向左平移一个单位，将问题转化为矩形的点权和减去梯形点权和。
- 令 $m=\left\lfloor\frac{an+b}{c}\right\rfloor$。那么矩形的点权和是 $S_q=\sum_{i=1}^{m}\sum_{j=1}^{n} i^q j^p$。注意，由于我们将平面沿 $y=x$ 翻转，因此点权是 $i^q j^p$。
- 梯形内的点权和是 $f_{c,c-b-1,a,m-1}$。对于这个，我们得做变换：$(i,j)\to(i+1,j)\to(j,i+1)$。点权变换是 $i^p j^q\to j^p(i+1)^q$。
- 点权变换仍可以用二项式定理展开。
- 于是 $f_{a,b,c,n}(p,q)=S_q+\sum_{t=0}^{q}\binom{q}{t}f_{c,c-b-1,a,m-1}(t,p)$。

A Math Challenge

牛客竞赛 AC.NOWCODER.COM

- 剩下一个问题：自然数幂和（等幂求和）怎么做。
- 有一个棘手的点：计算 Case 2 中的 $S_t = \sum_{i=0}^{n} \sum_{j=1}^{ki} i^p j^q$。
- 考虑求出自然数幂和的多项式的系数表示。设 $F_q(n) = \sum_{i=0}^{n} i^q = \sum_{i=0}^{q+1} c_i n^i$，不妨设 $q > 0$。
- $S_t = \sum_{i=0}^{n} i^p \sum_{t=0}^{q+1} c_t (ki)^t = \sum_{t=0}^{q+1} c_t k^t F_{p+t}(n)$。
- 这就转化为了自然数幂和的问题。
- 时间复杂度 $O\big((p+q)^3 \log \max(a,c)\big)$。

Best Subgraph

- 首先可以发现，如果不考虑连通性问题，(k+1)-degree 一定是 k-degree 的子图
- 如果一个点属于 (k+1)-degree 而不属于 k-degree，我们可以把这个点标记为 k+1
- 所以我们可以考虑找到 k 最大的 k-degree，然后考虑不断的拓展这个子图，去得到 k 比较小的 k-degree
- 显然每一次加入一批相同标记的点，可以拓展得到k减小的 k-degree

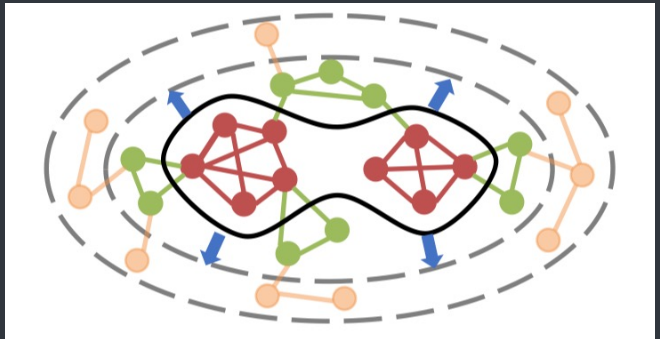
Best Subgraph

牛客竞赛
AC.NOWCODER.COM

- 假设我们已经知道了当前图的 score ，我们考虑新增点/边会带来的 score 变化
- 我们首先给所有的节点一个 rank
  - $rank(v) > rank(u)$ 当且仅当
    - v的标记 > u的标记
    - v的标记 = u的标记 且 $v > u$
  - 这个部分可以用 bin-sort 完成
- 我们可以根据边两边点的 rank 对边分类
  - 用 E(v,>) 表示 v 的边中另一个端点比 v rank 大的边
  - 其余的表示以此类推

Best Subgraph

- 于是，考虑新增一批点（标记相同的点）：
  - 对于每一个加入的点 v
    - $\Delta n = 1$
    - $\Delta m = |E(v, >)| + \frac{1}{2}|E(v, =)|$
    - $\Delta b = |E(v, <)| - |E(v, >)|$



- 于是从标记大的点到小的点依次加入计算 score，得到最大score的k即可
- 因为不断加入点，可能使得本来不连通的两个子图联通起来，这部分可以用并查集维护
- 时间复杂度 O(m+n)

Cells

- 首先根据 LGV 引理，我们所要求的就是$A=\begin{vmatrix} \binom{a_1+1}{1} & \binom{a_1+2}{2} & \cdots & \binom{a_1+n}{n} \\ \binom{a_2+1}{1} & \binom{a_2+2}{2} & \cdots & \binom{a_2+n}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \binom{a_n+1}{1} & \binom{a_n+2}{2} & \cdots & \binom{a_n+n}{n} \end{vmatrix}$
- 每一列提取$\frac{1}{j!}$后可以让每一行都变成a_i的j次多项式，即$A'_{i,j}=\frac{(a_i+j)!}{a_i!}=\prod_{k=1}^{j}(a_i+k)$
- 显然，每一列都变成一样的了

Cells

- 利用行列式初等变换，让靠前的列去消掉靠后的列，可以得到$A''_{i,j} = (a_i + 1)^j$
- 显然，这是一个范德蒙德行列式，也就是
  $$A'' = \prod (a_i + 1) \cdot \prod_{1 \le i < j \le n} ((a_j + 1) - (a_i + 1))$$
- 则有$A = \prod_{j=1}^{n} \frac{1}{j!} \cdot \prod (a_i + 1) \cdot \prod_{1 \le i < j \le n} ((a_j + 1) - (a_i + 1))$
- 计算 $\prod_{1 \le i < j \le n} ((a_j + 1) - (a_i + 1))$ 需要用卷积
- 时间复杂度 $O(n \log n)$

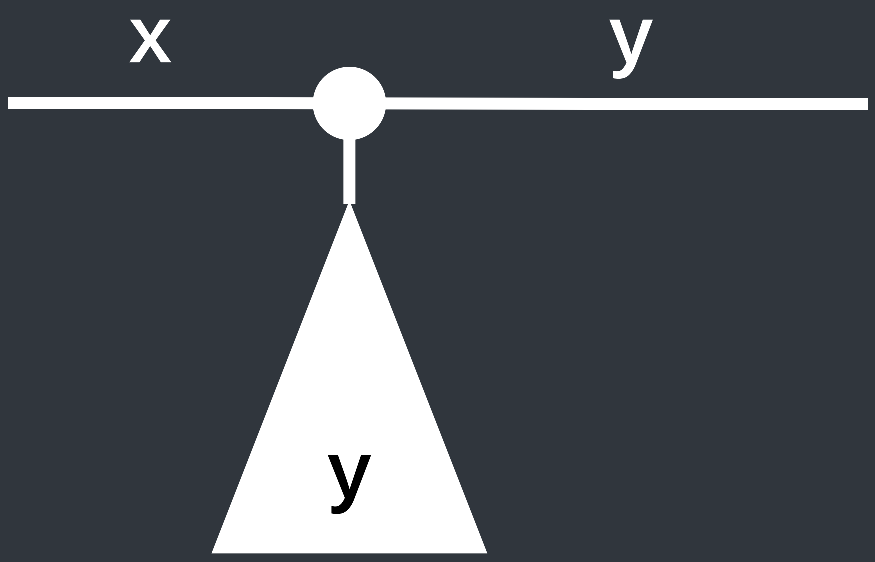
Divide-and-conquer on Tree

牛客竞赛
AC.NOWCODER.COM

- 题意：构造·一棵二叉树以及一种边分治的策略，使得迭代次数比按 子树大小 边分治的迭代次数至少少2层
- 思路：考虑按子树大小边分治与最少迭代次数的边分治的区别。
- 按子树大小分治可能会将树分为一棵迭代层数很小的子树和一棵迭代层数很小的子树。比如，菊花图上为迭代层数为n-1，而链上迭代层数为$\log_2 n$。

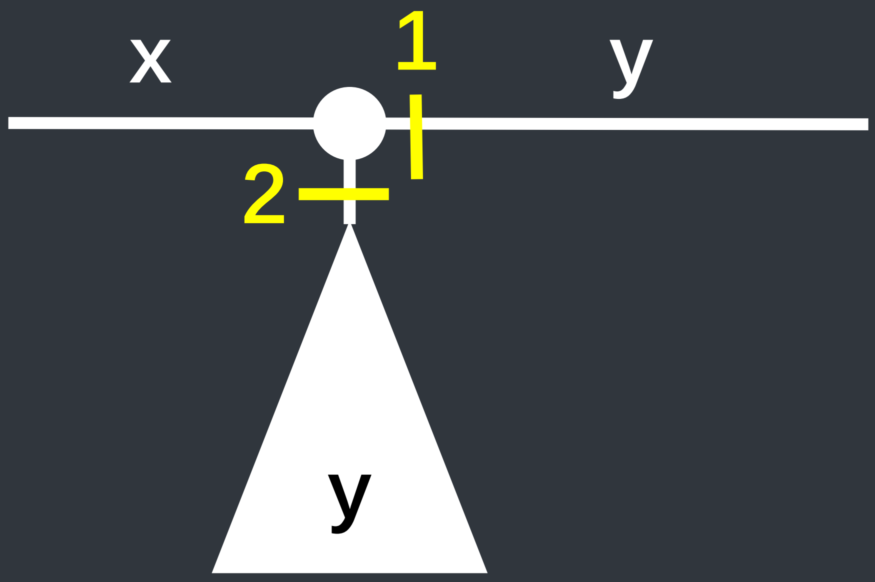
Divide-and-conquer on Tree

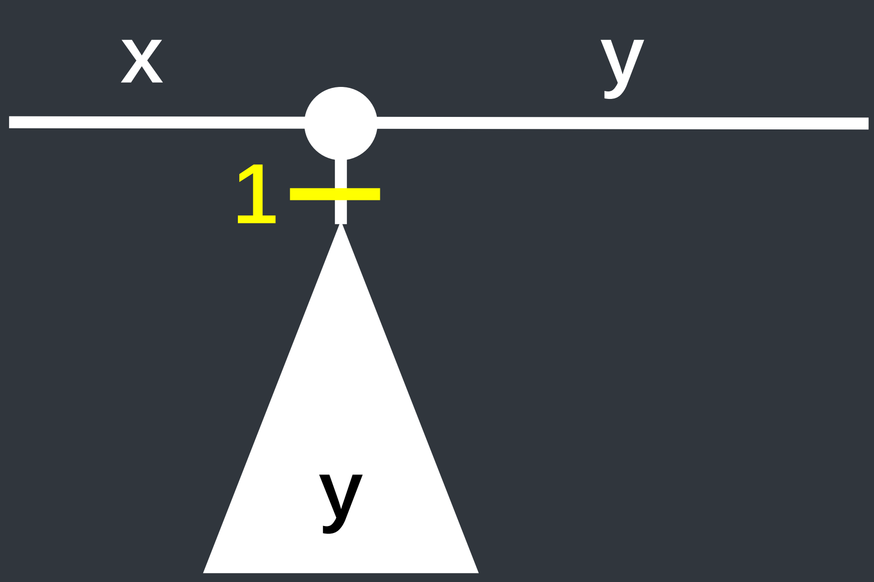
- 考虑二叉树的特点，每个点度数不大于3。观察以下例子：



Divide-and-conquer on Tree

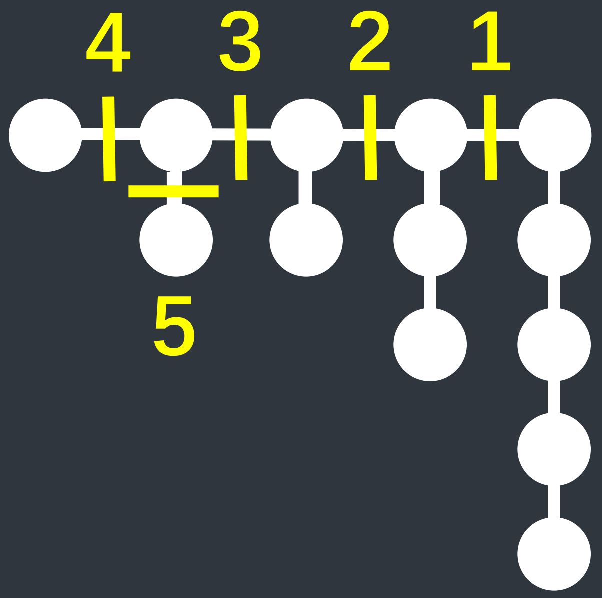
• 考虑二叉树的特点，每个点度数不大于3。观察以下例子：

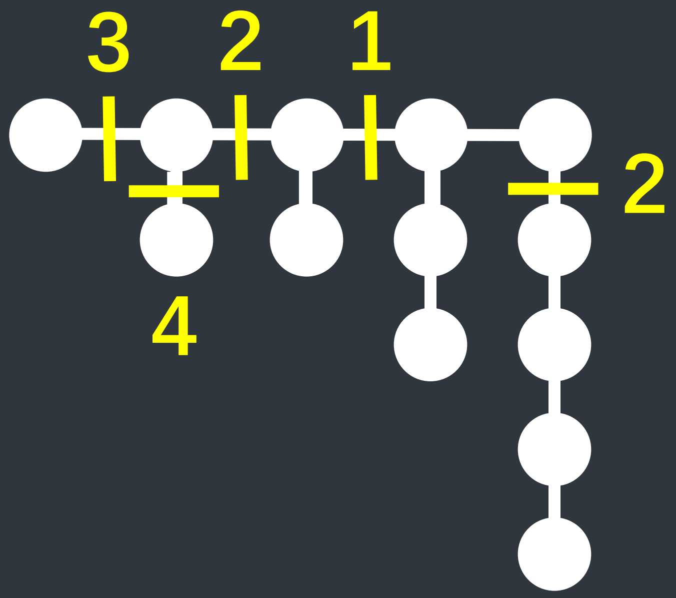




Divide-and-conquer on Tree

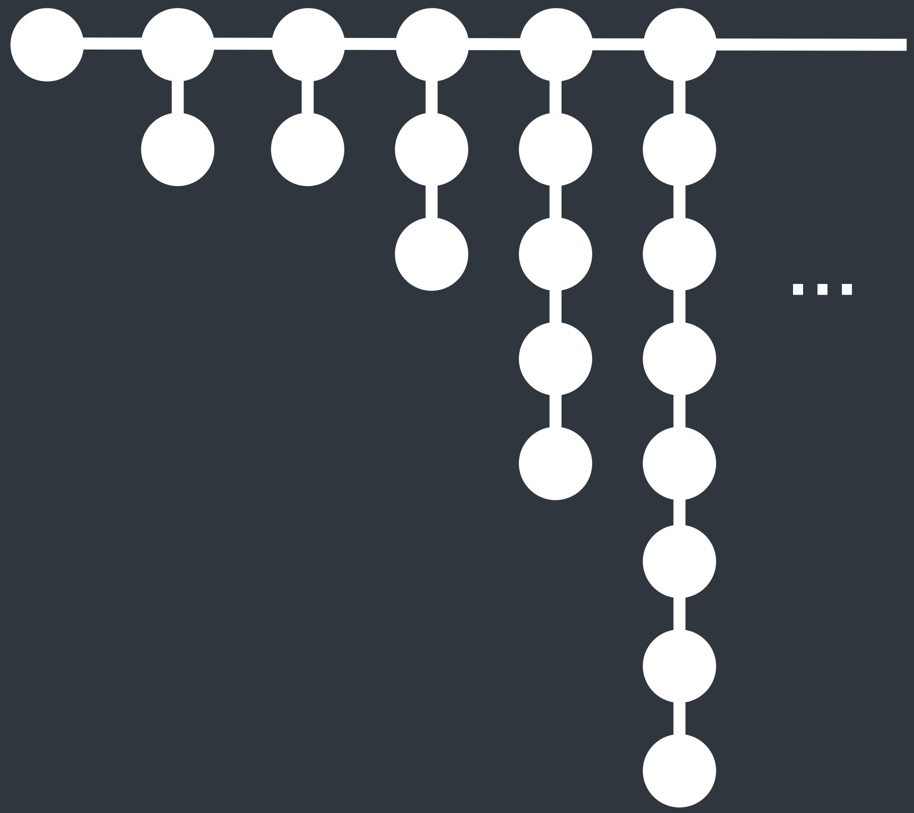
- 构造方法：





Divide-and-conquer on Tree

- 构造方法：



$$\begin{array}{ccccccccc} & & & & 2 & & 1 & & \\ 1 & 2 & 2 & 3 & 5 \,|\, 8 & 13 & |\, 21 & 34 \end{array}$$

$$\mathrm{Log}_2 x_{n+1} + 1 = 7$$

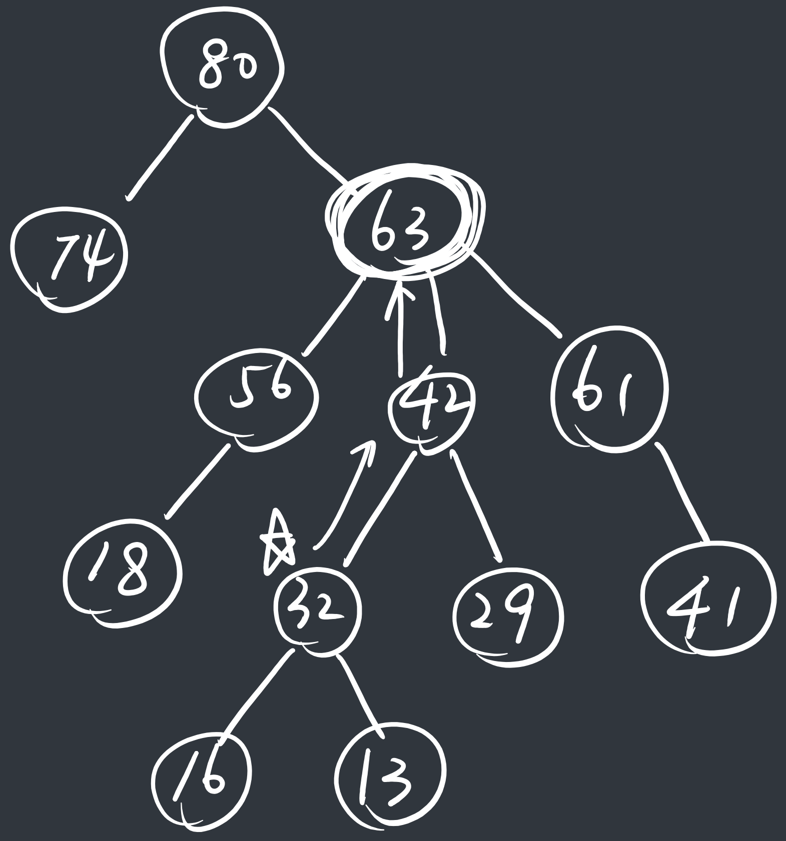
$$n = 9$$

E Eyjafjalla

- 题意：给定一个以 1 为根的有根树，孩子的点权小于父亲的点权。多次询问，每次询问包含 x 节点的权值范围为 $[l, r]$ 的极大连通块的大小。

E Eyjafjalla

- 病毒传播可以看作两个阶段，第一个阶段先上升到最高的一个节点 p，第二阶段感染 p 的子树中所有温度大于 l 的城市。
- 第一阶段可以通过倍增法求得 p；
- 第二阶段相当于在 p 的子树中查询权值大于 l 的节点个数，根据每个节点的 dfs 序建立可持久化线段树，然后在线段树上查询即可。
- 时间复杂度 $O(n \log n)$。



[20, 70]

Financial Order Execution

- 题意：一天要买/卖若干股票，要求把订单拆到分钟级别。总共有 240 分钟，第 i 分钟成交价格为 p[i]，最多成交 v[i] 股。另有交易单位（一手 100 股），佣金（万分之三，最少五元）的限制。求最少花费/最大收益。
- 首先我们将所有计算单位转化成分，成交量单位转化为手。
- 考虑 dp，f(i, j) 表示到第 i 分钟，成交 j 手，最大收益（考虑买单，卖单同理）。

$$f(i,j)=\min_{0\le j-k\le v_i}\left\{f(i-1,k)+100(j-k)\cdot p_i-\max\left\{\left\lceil (j-k)\cdot p_i\cdot\frac{3}{100}\right\rceil,500\right\}\right\}$$

Financial Order Execution

- 将 max 打开并整理 j 和 k 得到（不考虑 $j = k$）

$$f^*(i,j) = \min_{0\le j-k\le v_i} \begin{cases} f(i-1,k) - 100p_ik + (100p_ij - 500), & 3p_i(j-k) \le 50000 \\ f(i-1,k) - 100p_ik + \dfrac{3}{100}p_ik + (100p_ij - \dfrac{3}{100}p_ij), & 3p_i(j-k) > 50000 \end{cases}$$

- 注意在分支 2 中本来有一个取整函数，但是若我们只求最优的 k，我们无须关心取整，因为取整是单调的。
- 求出最优 k 后代入原式即可求出最优的 f(i, j)。
- 显然该式可以用两个单调队列优化，复杂度可以做到 $O(nV)$。

Glass Balls

- 首先转化题意，本题实际上是要求将树划分成若干条祖孙方向的链，一条长度为 k 的链的价值是 $\frac{k(k-1)}{2}$，求总价值的期望。
- 考虑其组合意义，相当于是在划分为若干条链后，每条链上各选两个节点的方案数。
- 考虑 DP，设 $f(i,j)$ 表示考虑节点 i 的子树，且 i 所在的链已经选了 j 个点的方案数。转移不难。

Happy Number

- 签到题

- 首先需要确定出 k 大数有多少位
  - 显然1位的、2位的…分别有$3, 3^2, \cdots$
- 于是我们知道了 k 大数是 x 位下第 k′ 大的
- 然后我们可以把 2 看做 0，3 看做 1 ，6 看做 2
- 问题就转换成了三进制数转换

Incentive Model

- 令 $X_i \in \{0,1\}$分别表示矿工 A 能不能获得第 i 个区块
  - X_i 满足伯努利分布
- 令 S_i 表示在竞争完第 i 个区块后 A 的 stake
  - 显然 $S_0 = a$
- 所以 X_i 为 1 的概率是 $\dfrac{S_{i-1}}{1 + w(i-1)}$（已经发放了 i-1 次奖励）
- 显然 $S_{i+1} = S_i + wX_{i+1}$

Incentive Model

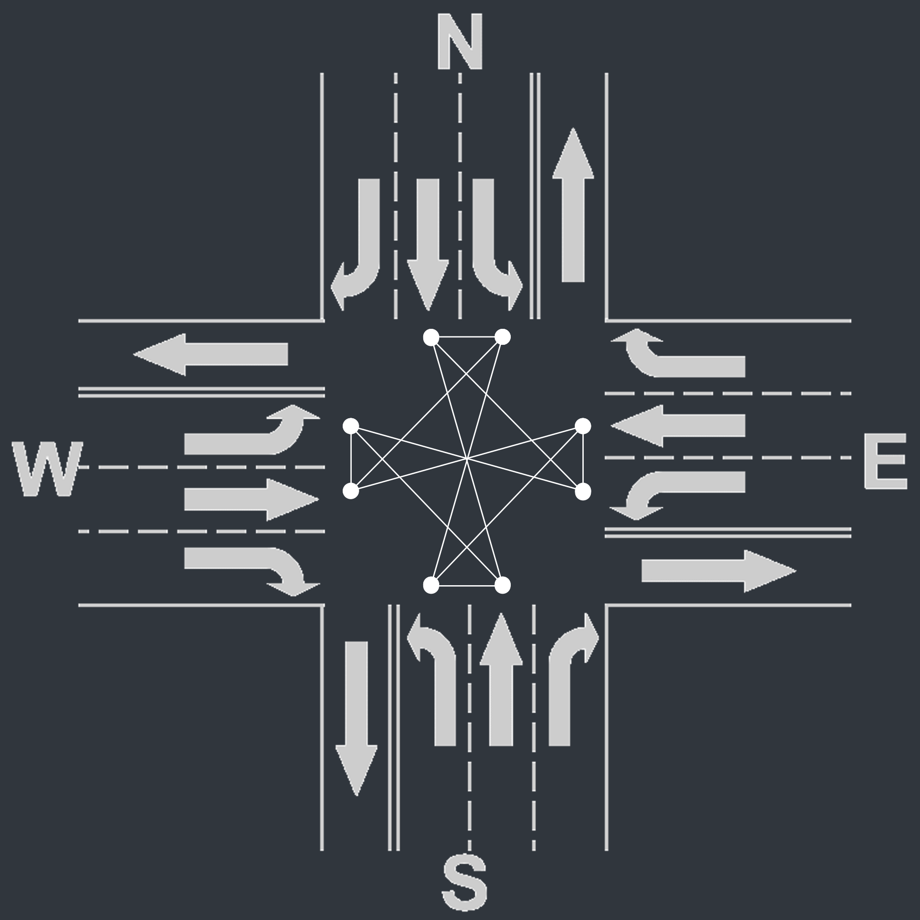
- 于是我们有 $\mathbb{E}[S_{i+1} \mid S_i] = S_i + \dfrac{wS_i}{1+wi}$
- 所以

$$\mathbb{E}[S_{i+1}] = \mathbb{E}[\mathbb{E}[S_{i+1} \mid S_i]] = \frac{1+w(i+1)}{1+wi} \cdot \mathbb{E}[S_i] = S_0 \prod_{k=0}^{i-1} \frac{1+w(k+1)}{1+wk} = a(1+wi)$$

- 于是$\mathbb{E}[\lambda_A] = \dfrac{\mathbb{E}[S_n] - a}{wn} = a$

Jam

- 首先右转永远都能走。
- 剩下的可以两两组合一起走，总共十 12 种组合。
- 将组合两两相连，得到右图。

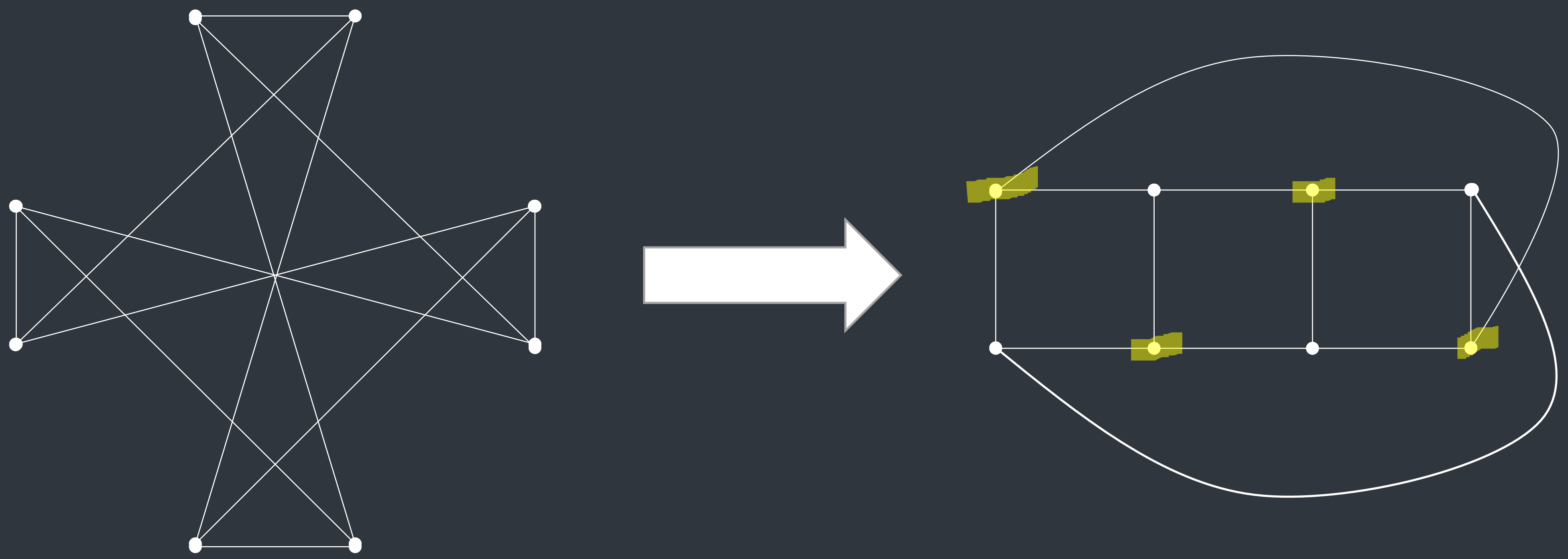


Jam

- 问题转化成每条点上有个最大匹配量，求整个图的最大匹配数。
- 可以尝试硬解：
  - 线性规划
  - 一般图匹配
  - ……
- 不一定能过。
- 如果是二分图就好了……

Jam

- 重新画下图。



Jam

- 这个图差两条多出来的边就是二分图了。
- 枚举多出来的两条边的匹配数，然后无视这两条边，其余边用二分图匹配就行了。
- 复杂度是 $O(n^2C)$。其中 C 是网络流的复杂度。

- 还有很多别的方法……