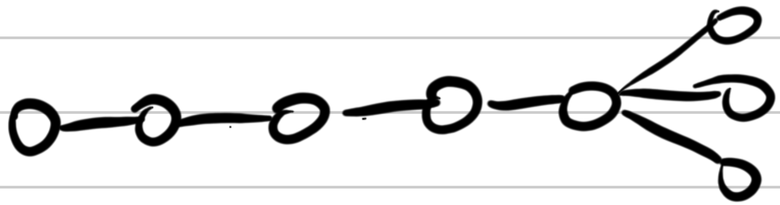
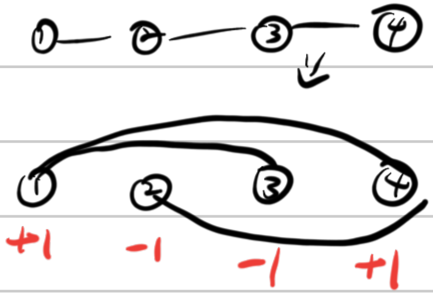
A

① 首先我希望变成形如

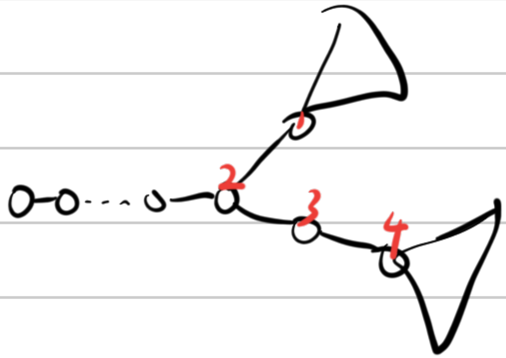


这样的状态;

对一次操作考虑度数变化




对于

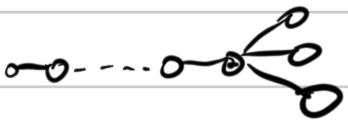


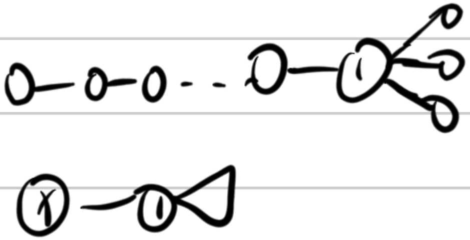
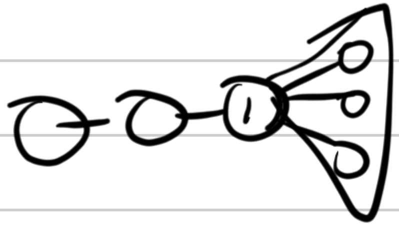
操作 1,2,3,4. 可使 ②度数 -1

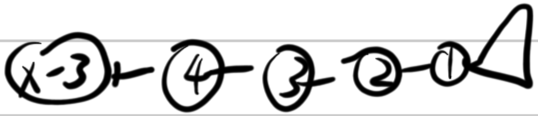
直到 ①度数为2,

→ 做下一个点.

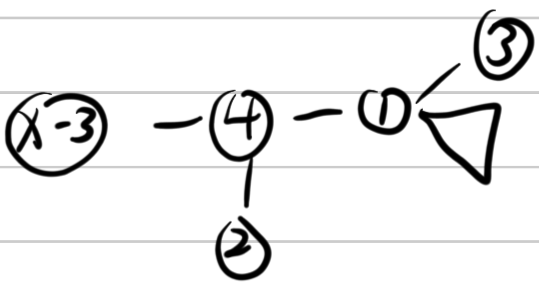
② 形如

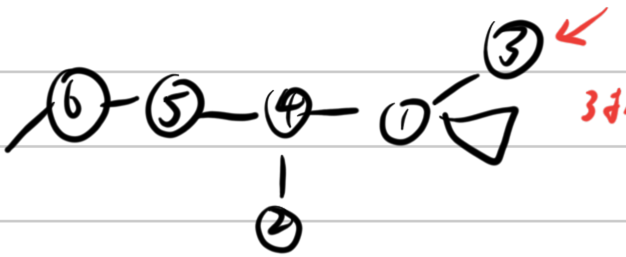


② 把  变成 

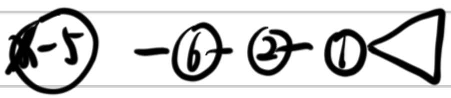
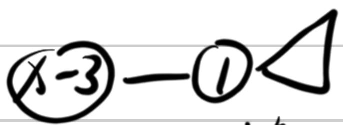


→ opt(1, 2, 3, 4)

⇒  若 x-3 ≤ 1 end

若 x-3 ≥ 2  3折叠进去

opt(1 4 5 6) opt(1 6 4 2)

⇒  ⇔  重复

状态.

代码 2K. 20min

# J

1. 仅有最后一步需要"走近"，其它 extend 都更优。（同颜色之间转移无意义）

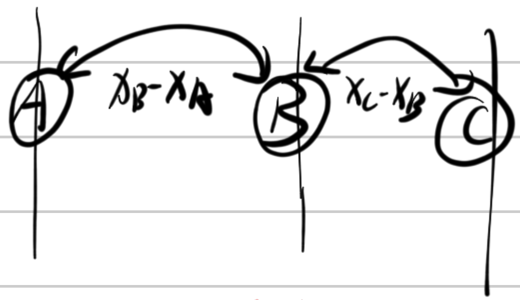
2. 建图，dist[i]：到 i 的距离

考虑走 x 轴，之后 extend

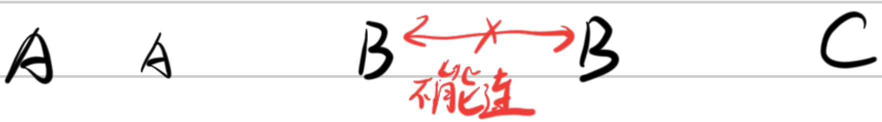
连边，$d[i][j] = abs(x_i - x_j)$　　$c_i \neq c_j$

$n^2$ 条边

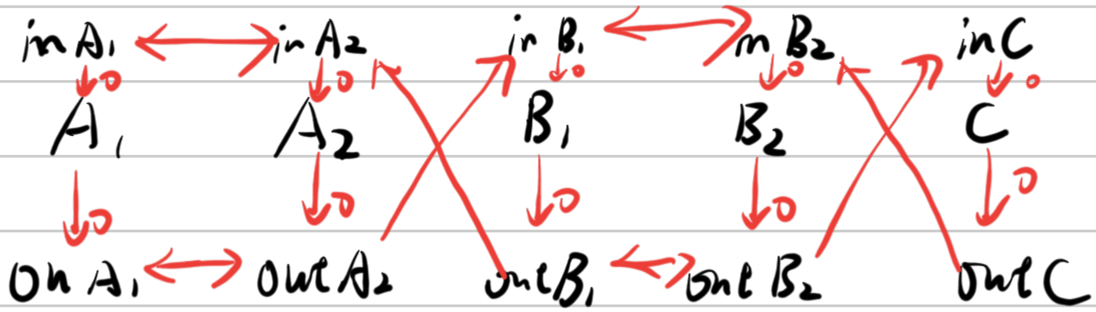
满足传递性，只须用按 x 前后排序，连前后继 $O(n)$



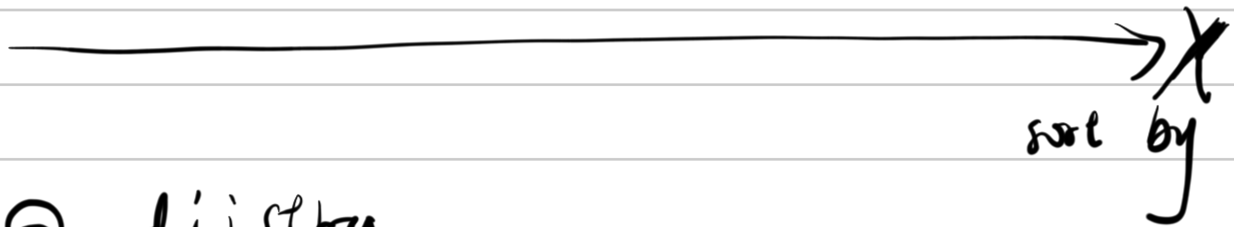

但还需 $c_i \neq c_j$ 的条件：




新建 in(id)，out(id) 结点，表示只能连和自己颜色不同的入边/出边




可以验证这样建图可以保证同颜色相邻点没有$|x_j - x_i|$的转移边




③ dijstra

代码2.5K以内

K：直接考虑更普遍的模型：

☆ n条格子，格子上有数字（可以负数），两个轮流选一条格子并走一步，获得分数，问先手能拿几分

$\rightarrow \boxed{3}\ \boxed{-1}\ \boxed{4}$　　先手能拿10分，后手0分，

$\rightarrow \boxed{1}\ \boxed{4}$

① 性质1：对于一条格子的最后两个数，

若 $a_{n-1} < a_n$，可删除这两个格子，

若总格子为奇数，则先手 $+a_{n-1}$，后手 $+a_n$，

反之同理。

自证

② 假设所有格子 $a_i \geq a_{i-1}$:

尝试在某个格子开头新增一个格子

4 10 7 4 2 −10

断言: 先手选了4, 后手一定选10, 先手必然选7
$a_1$ $a_2$ $a_3$

若此时选其它格子, 不如直接第一步就选, 就不用亏6分了

故, 上式等价于 $(a_1 a_2 a_3) = a_1 - a_2 + a_3$ 先手比后手多的分数

$\Leftrightarrow$ [1] 4 2 −10 $\Leftrightarrow$ [−1] −10

约化后 格子都单减

③ 直接所有格子 sort, 前手拿全局最大

L

代价 $= \sum A + \frac{rC}{\sum B}$.

记 $f[A] = \max B$

$A$ 值域 $\le 300$, ($n \le 300$).

完全背包的一个性质: $f[\sum A] = \max B$.

的购买方案, 最多只有一种物品数量大于 $\max A_i$

处理出 $V = \max A_i$, $V^2$ 以内的 $f(A)$.

枚举 $F[V^2]$: $F[A] = B$

再枚举物品 $i$, 所有方案都可表示为 $A + kA_i \Rightarrow B + kB_i$

使得 $\sum A + \frac{rC}{\sum B}$ 最小的 $k$ $O(1)$ 计算