

Notes

A. Stop, Yesterday Please No More

袋鼠题. 相对运动, 反向移动边界和洞.

首先移动边界, 得到剩余袋鼠所在范围. 接着模拟移动洞的路径, 利用二维前缀和枚举洞所在位置.

能够计算二维前缀和的前提是, 移动边界之后, 如果有袋鼠残留, 说明洞的轨迹大概是在 $n \times m$ 的范围内, 稍微开大一点空间计算二维前缀即可.

B. Ropeway

滑索. 单调队列优化 dp.

先考虑不修改, 就是一个单调队列优化 dp, 注意边界, 先将 0 放入队列, 接着要跑到 $n + 1$ 的位置, 对于必须要选择的位置只需要将队列清空再将这类点放入队列即可.

然后考虑修改, 假设修改的位置是 pos , 实际上发生改变的值只有 pos 到 $n + 1$ 这一段, 但是我们不能又跑一个 pos 到 $n + 1$ 这一段的 dp.

注意到 k 的限制, 如果我们先跑一个后缀的单调队列优化 dp, 记为 g , 这样即使 pos 位置的值发生改变, 导致 $[pos, pos + k]$ 这一段的 dp 值改变, 我们也可以枚举 $x \in [pos + 1, pos + k + 1]$, 强制选择这个位置, 答案对 $g(x) + \min_{y < x, x - y \leq k} \{f_{new}(y)\}$ 取 min 即可.

至于这一段的实现, 首先需要得到新的 dp 值 f_{new} , 先拷贝向前 k 的答案, 再往后算 k 步, 接着更新答案, 更新的过程也是一个单调队列优化 dp.

所以这个题是 4 个单调队列优化 dp.

D. Chat Program

一个序列加上一个固定长度的等差数列, 求第 k 大数的最大值. 二分答案.

考虑二分答案, 对于每一个答案 x , 只需要知道是否有一种的方式使得序列加上等差序列之后, 有至少 k 个数不小于 x .

这个可以通过枚举每一个 $i \in [1, n]$, 计算能使 a_i 加上等差数列中的某个值使其不小于 k 的等差数列的位置. 这只需要记录等差数列中的第一个数的位置即可. 然后实际上就是考虑是否有一个位置的值大于等于 k .

考虑到等差数列的单调性, 可以分成如下几类:

1. 本身就不小于 x , 可行的位置为 $[1, n - m + 1]$;
2. 即使加上最大值都不够, 没有可行的位置;
3. 加上最小的就够了, 可行的位置为 $[i - ri + 1, i - le + 1]$;
4. 直接计算出满足 $a_i + c + d(t - 1) \geq x$ 的最小的 t , 可行的位置为 $[i - ri + 1, i - t + 1]$.

等差数列的单调性保证可行的位置一定是一个连续的区间, 没必要写线段树区间修改之类的, 一个差分在前缀和之后求最大值即可.

E. Color the Tree

将一棵树上的点染黑, 每次只能染子树中同层的点. 虚树加树形 dp.

对整棵树而言, 完全可以一层一层考虑.

对第 i 层 $layer$ 而言, 其深度为 d . 将这些点建立一棵虚树, 考虑 dp 转移式如下

$$f_u = \min \left(f_u + \sum_{v \text{ is son of } u} f_v \right),$$

也就是考虑是否用 u 染叶子.

然后就要考虑如果用的代价是多少. 这里就不再是在原树中的 $dep_u - d$ 了, 而是 $\min_{d - dep_v \leq i \leq d - dep_u - 1} \{a_i\}$, 预处理一个 ST 表即可.

I. Perfect Palindrome

判断字符串是否每一个后缀回文. 签.

实际上就是判断是否所有字符相同.

J. Perfect Matching

满足 $|i - j| = |a_i - a_j|$ 的 i, j 之间有连边, 求完美匹配. dfs.

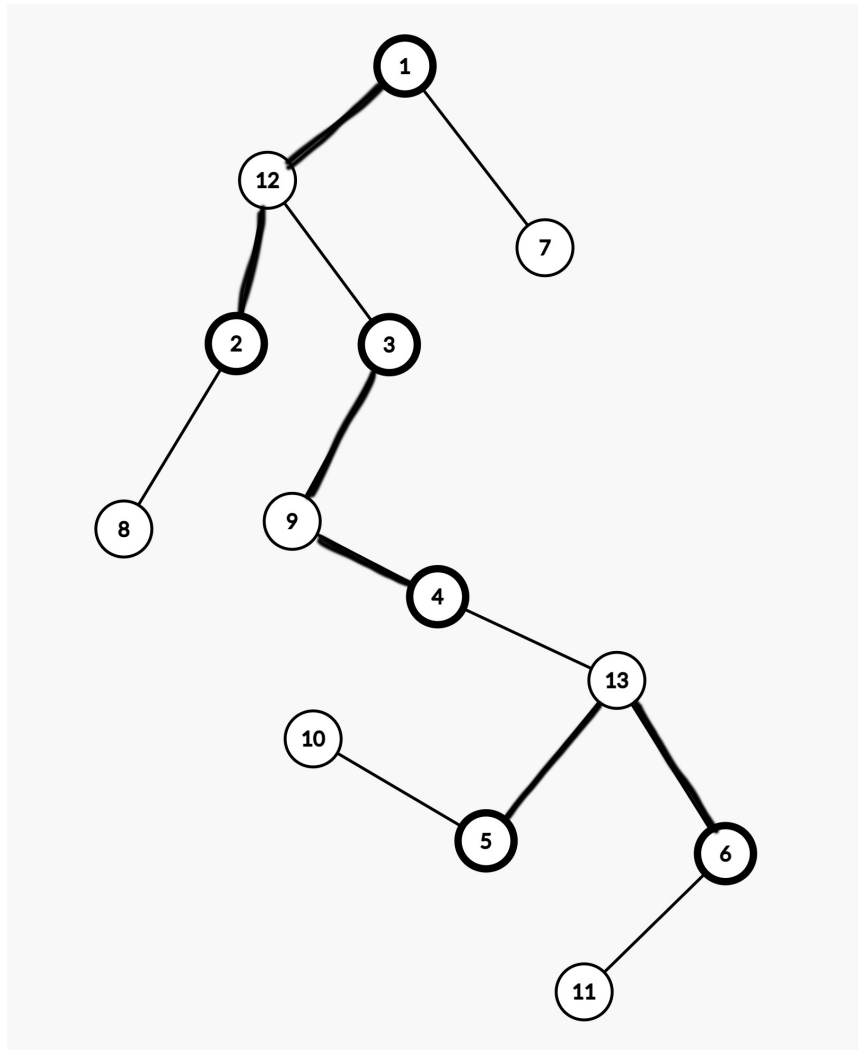
转换题意, 上式等价于 $a_i + i = a_j + j$ 或者 $a_i - i = a_j - j$. 如果把所有 $a_i + i$ 和 $a_j + j$ 算出来离散化, 将 i 与他们俩连边这这就是一个二分图, 接着就是在二分图中将 $i, j \leq n$ 通过长度为 2 的链两两匹配.

有点贪心的思路, 对于每一个连通块, 值不超过为 n 的点的数量为偶数一定构造出一组解, 否则这一组样例无界.

接着开始贪心, 对于 $u > n$ 而言, 与它相连的是需要匹配的点, 将他们都放入集合 ver , 接着将 ver 内的点两两匹配, 如果是奇数个, 将剩下的与 u 的父亲匹配即可.

例如数据为

```
6
2 3 4 3 4 5
```

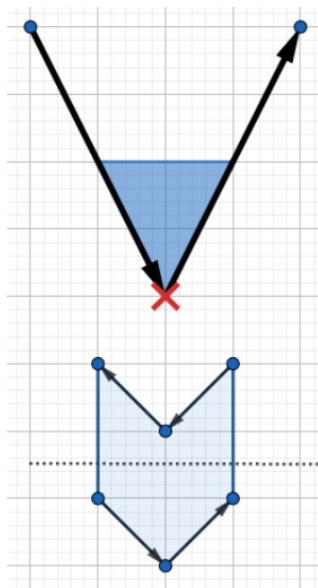


M. Drain the Water Tank

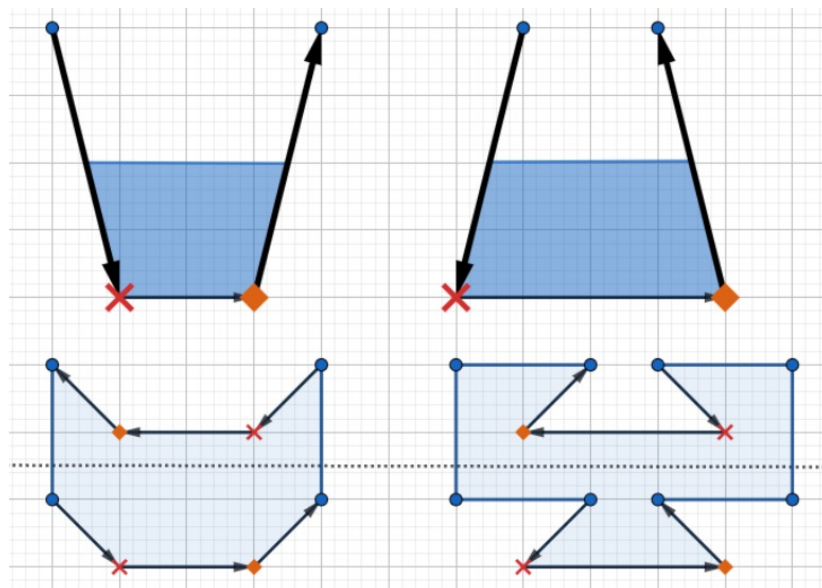
给水箱找出水口. 初级计算几何?

找局部最低点. 有三种情况:

1. 上一个点的纵坐标比当前大, 下一个点的纵坐标比当前小, 判断一下外积, 不然会计算天花板上的点;



2. 上一个点的纵坐标比当前大, 下一个点的纵坐标和当前一样, 找到下一个纵坐标和当前不一样的点, 判断是否纵坐标变大了, 不需要判断外积.



不这样算总能找到 coner case.