

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

杂题选讲

吴清月

2020 年 7 月 27 日

Trivial

●○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

联合权值

【NOIP2014】联合权值

【NOIP2014】联合权值

一棵 n 个点的树，每一个点有一个点权。

若两个点距离为 2，那么它们就会产生它们的点权之积的联合权值。

求整棵树的联合权值之和。

$$1 \leq n \leq 200000$$

Trivial



Easy



Medium



Not Hard



Hard



联合权值

【NOIP2014】联合权值

Trivial	Easy	Medium	Not Hard	Hard
○● ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○
联合权值				

【NOIP2014】联合权值

若距离为 2，则一定存在一个中转点。
 枚举这个中转点，周围点权之积可以直接算。
 时间复杂度 $O(n)$ 。

食物链

【NOI2001】食物链

【NOI2001】食物链

有三种生物 A,B,C, A 吃 B, B 吃 C, C 吃 A。

现在有 n 个生物, 还有 k 个条件, 每一个条件形如下面一种:

- X 和 Y 是同类。
- X 吃 Y 。

问你有多少句话和前面的话冲突。

$$n \leq 5 \times 10^5, k \leq 10^5$$

Trivial

○○
○●
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

食物链

【NOI2001】食物链

Trivial

○○
○●
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

食物链

【NOI2001】食物链

在关押罪犯那道题中，我们用 X 和 X' 表示是否和自己在同一间监狱里面。

在这道题中，我们就设 X_1, X_2, X_3 分别表示自己，食物和天敌。

Trivial

○○
○●
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

食物链

【NOI2001】食物链

在关押罪犯那道题中，我们用 X 和 X' 表示是否和自己在同一间监狱里面。

在这道题中，我们就设 X_1, X_2, X_3 分别表示自己，食物和天敌。

用并查集维护，如果在任何时刻 X_1, X_2, X_3 在同一个集合中那就不合法。

时间复杂度 $O(k\alpha(n))$ 。

Trivial



Easy



Medium



Not Hard



Hard



车站分级

【NOIP2013】车站分级

【NOIP2013】车站分级

一条单向的铁路线上，依次有编号为 $1, 2, \dots, n$ 的 n 个火车站。每个火车站都有一个级别，最低为 1 级。

现有若干趟车次在这条线路上行驶，每一趟都满足如下要求：

如果这趟车次停靠了火车站 x ，则始发站、终点站之间所有级别大于等于火车站 x 的都必须停靠。（注意：起始站和终点站自然也算作事先已知需要停靠的站点）

现有 m 趟车次的运行情况（全部满足要求），问这 n 个火车站至少分为几个不同的级别。

$$n, m \leq 1000$$

Trivial

○○
○○
●○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

车站分级

【NOIP2013】车站分级

Trivial

○○
○○
●

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

车站分级

【NOIP2013】车站分级

对于一列火车，停靠的所有车站级别一定大于未停靠的，相当于拓扑排序中的偏序关系。由于一定满足条件，因此一定能得到一个有向无环图。

拓扑排序的时候顺带进行 DP，设 f_i 表示 i 的最小级别。

Trivial

○○
○○
●

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

车站分级

【NOIP2013】车站分级

对于一列火车，停靠的所有车站级别一定大于未停靠的，相当于拓扑排序中的偏序关系。由于一定满足条件，因此一定能得到一个有向无环图。

拓扑排序的时候顺带进行 DP，设 f_i 表示 i 的最小级别。直接暴力建图，时间复杂度 $O(mn^2)$ 。

【NOIP2013】车站分级

对于一列火车，停靠的所有车站级别一定大于未停靠的，相当于拓扑排序中的偏序关系。由于一定满足条件，因此一定能得到一个有向无环图。

拓扑排序的时候顺带进行 DP，设 f_i 表示 i 的最小级别。

直接暴力建图，时间复杂度 $O(mn^2)$ 。

可以引入辅助点，时间复杂度降到 $O(nm)$ 。

○ ○

○ ○

○ ○

BZOJ 2563

Trivial

○○
○○
○○

Easy

●○
○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○
○○○○○

阿狸和桃子的游戏

BZOJ 2563

有一张 n 个点 m 条边的图，点有点权，边有边权。

先手后手轮流染黑白两色，最后的得分是自己染的点权和 + 两端均为自己的颜色的边权和。

双方都希望自己的得分-对手的得分最大，求结果。

$1 \leq n \leq 10000, 0 \leq m \leq 100000$



○ ●
○ ○
○ ○
○ ○

BZOJ 2563

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○● ○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

阿狸和桃子的游戏

BZOJ 2563

把一个点 u 的点权变为 $2w(u) + \sum w(u, v)$, 边权变为 0。
接下来贪心选取。

BZOJ 2563

把一个点 u 的点权变为 $2w(u) + \sum w(u, v)$ ，边权变为 0。

接下来贪心选取。

可以发现同色边权不会发生变化，异色边权两方分数可以相互抵消。

时间复杂度 $O(n \log n)$ 。

○○○

○ ○
● ○
○ ○
○ ○

A 4x3 grid of circles. The first three rows are complete, each containing 3 circles. The fourth row contains 2 circles, with the bottom-right position empty.

○ ○ ○ ○
○ ○ ○
○ ○ ○ ○ ○

运输计划

【NOIP2015】运输计划

Trivial

○○
○○
○○

Easy

○○
●○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

运输计划

【NOIP2015】运输计划

一棵 n 个点的树，给你 m 条路径。

你需要选出一条边把边权改成 0，让这些路径长度的最大值最小。

$$n, m \leq 3 \times 10^5$$

○ ○

○ ○

○ ○

运输计划

【NOIP2015】运输计划

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○● ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

运输计划

【NOIP2015】运输计划

最大值最小，先二分。

二分之后，有一些路径的长度已经满足要求了，直接忽略。

剩下的路径由于太长，所以每一条路径都必须经过被删除的边。

【NOIP2015】 运输计划

最大值最小，先二分。

二分之后，有一些路径的长度已经满足要求了，直接忽略。

剩下的路径由于太长，所以每一条路径都必须经过被删除的边。

假设有 k 条超过限制的路径，对于每一条路径，我们把这条路径上所有边的 $\text{cnt}+1$ ，然后看被所有 $\text{cnt}=k$ 的边，删掉其中最长的边，看是否能满足条件。

对于 $\text{cnt}+1$ 的操作，我们可以用树上差分实现。每次让 $\text{cnt}[u]++$, $\text{cnt}[v]++$, $\text{cnt}[\text{lca}]-=2$ 。

【NOIP2015】运输计划

最大值最小，先二分。

二分之后，有一些路径的长度已经满足要求了，直接忽略。

剩下的路径由于太长，所以每一条路径都必须经过被删除的边。

假设有 k 条超过限制的路径，对于每一条路径，我们把这条路径上所有边的 $\text{cnt}+1$ ，然后看被所有 $\text{cnt}=k$ 的边，删掉其中最长的边，看是否能满足条件。

对于 $\text{cnt}+1$ 的操作，我们可以用树上差分实现。每次让 $\text{cnt}[u]++$, $\text{cnt}[v]++$, $\text{cnt}[\text{lca}]-=2$ 。

时间复杂度 $O((n+m)\log n)$ ，有点卡常。

○ ○

○ ○

○ ○

○ ○
○ ○
● ○
○ ○

Orac and Game of Life

CodeForces 1349C

CodeForces 1349C

有一个 $n \times m$ 的网格，每一个格子初始是黑或白两种颜色。

接下来的每一个时刻，对于一个格子 (i, j) ：

- 若相邻的格子中有至少一个格子和它的颜色相同，那么这个格子的颜色取反。
- 否则，颜色不变。

q 次询问，每次问你 (x, y) 在第 t 时刻的颜色。

$1 \leq n, m \leq 1000, 1 \leq q \leq 100000, 1 \leq t \leq 10^{18}$

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○●	○○○	○○○	○○○○○
	○○	○○○	○○	

Orac and Game of Life

CodeForces 1349C

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	●○	○○○	○○○	○○○○○
	○○	○○○	○○	

Orac and Game of Life

CodeForces 1349C

首先我们找规律。

注意到，一旦某个连通块的大小超过 1，那么它就会无限 01 反转下去。

而如果周围有大小为 1 的连通块，那么这两个连通块就会合并，然后继续一起反转。

CodeForces 1349C

首先我们找规律。

注意到，一旦某个连通块的大小超过 1，那么它就会无限 01 反转下去。

而如果周围有大小为 1 的连通块，那么这两个连通块就会合并，然后继续一起反转。

对于每一个格子，我们预处理一个 $c_{i,j}$ ，表示这个位置在这个时刻就会进入一个大小大于 1 的连通块。

对于询问, 如果 $t \leq c_{i,j}$ 那就不变, 否则就黑白交错出现。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○●	○○○	○○○	○○○○○
	○○	○○○	○○	

Orac and Game of Life

CodeForces 1349C

首先我们找规律。

注意到，一旦某个连通块的大小超过 1，那么它就会无限 01 反转下去。

而如果周围有大小为 1 的连通块，那么这两个连通块就会合并，然后继续一起反转。

对于每一个格子，我们预处理一个 $c_{i,j}$ ，表示这个位置在这个时刻就会进入一个大小大于 1 的连通块。

对于询问，如果 $t \leq c_{i,j}$ 那就不变，否则就黑白交错出现。

$c_{i,j}$ 的求法比较简单，建图，然后跑一遍最短路即可。

由于边权都是 1，所以可以直接 bfs。时间复杂度 $O(nm + q)$ 。

Trivial

○○
○○
○○

Easy

○○
○○
○○
●○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

Trails and Glades

CodeForces 209C

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ●○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Trails and Glades

CodeForces 209C

给一张图，添加最少的边使得整张图存在一条欧拉回路。

$$1 \leq n \leq 10^6, 0 \leq m \leq 10^6$$

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○ ●○	○○○ ○○ ○○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○○	○○○○ ○○○ ○○○○○

Trails and Glades

CodeForces 209C

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○ ●●	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Trails and Glades

CodeForces 209C

如果只有一个连通块，那就直接把度数为奇数的点两两相连。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○ ●	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Trails and Glades

CodeForces 209C

如果只有一个连通块，那就直接把度数为奇数的点两两相连。

否则如果有 k 个连通块，我们需要先花费 $k - 1$ 条边把所有连通块连接。

优先选取度数为奇数的点向外连。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	●○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Construct the Binary Tree

CodeForces 1311E

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	●○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Construct the Binary Tree

CodeForces 1311E

给定 n 和 d , 你需要构造一棵 n 个点的二叉树, 满足所有点的深度之和恰好为 d 。

$$2 \leq n, d \leq 5000$$

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Construct the Binary Tree

CodeForces 1311E

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Construct the Binary Tree

CodeForces 1311E

首先考虑深度最小的情况，也就是完全二叉树。如果 d 小于这个数直接不可能。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Construct the Binary Tree

CodeForces 1311E

首先考虑深度最小的情况，也就是完全二叉树。如果 d 小于这个数直接不可能。

否则，我们考虑对这棵树进行修改，每次让总深度 $+1$ 。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Construct the Binary Tree

CodeForces 1311E

首先考虑深度最小的情况，也就是完全二叉树。如果 d 小于这个数直接不可能。

否则，我们考虑对这棵树进行修改，每次让总深度 $+1$ 。

首先，随便拎出来一条最深的链，为了方便我们直接把 1 到 n 拎出来。

然后我们开始倒序考虑不在这条链上的所有点，每次尝试把这个点的深度 $+1$ 。

只需要让这个点的父节点变为链上深度等于它的节点即可。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Construct the Binary Tree

CodeForces 1311E

首先考虑深度最小的情况，也就是完全二叉树。如果 d 小于这个数直接不可能。

否则，我们考虑对这棵树进行修改，每次让总深度 $+1$ 。

首先，随便拎出来一条最深的链，为了方便我们直接把 1 到 n 拎出来。

然后我们开始倒序考虑不在这条链上的所有点，每次尝试把这个点的深度 $+1$ 。

只需要让这个点的父节点变为链上深度等于它的节点即可。

每次我们刚好把总深度 $+1$ ，所以一定能够得到所有可行的结果。如果最后还没达到 d 那就不可能。

时间复杂度 $O(n + d)$ 。







Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○● ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Construct the Binary Tree

CodeForces 1311E

Construct the Binary Tree

CodeForces 1311E

		I got it..and Thanks for helping!!	Has been sent 5 months ago Has been read 5 months ago Reply
		I hope I can explain it clearly. Firstly, we build a complete balanced tree, in which we have $f[i]=i/2$ for $i>1$. We can see in this way we minimize the total depth, so if d is less than it, we can print "NO" and quit. Then, let's see the chain from n to 1 , and for node x on this chain, let $flag[x]=1$ and $node[depth[x]]=x$. For i from n to 1 , we see if node i is on the chain. If so we do nothing, otherwise we change its father to the node on the chain whose depth is 1 bigger. Keep doing this until i becomes the deepest node or current total depth is equal to d given in the input. Since i 's children has already been removed, each time we add exactly 1 to the total depth, so we will reach d anyway.(Unless d is too large to be reached)	Has been sent 5 months ago Has been read 5 months ago Reply
		Hello , Can you please explain me logic of "E.Construct the Binary Tree" from Codeforces Round #624 (Div 3)?	Has been sent 5 months ago Has been read 5 months ago Reply

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
●○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

华容道

【NOIP2013】华容道

【NOIP2013】华容道

有一个 $n \times m$ 的棋盘，其中有若干个位置是障碍，还有一个空白位置，其它位置都是棋子。

有 q 组询问，每次告诉你一个指定的棋子 (SX, SY) ，空白位置 (EX, EY) 和目标位置 (TX, TY) ，你需要求出将指定的棋子移动到目标位置最少需要移动几次。

$$1 \leq n, m \leq 30, 1 \leq q \leq 500$$

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○●
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

华容道

【NOIP2013】华容道

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○● ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

华容道

【NOIP2013】华容道

首先我们有一个暴力的想法。设 $f(i, j, k, l)$ 表示指定棋子位于 (i, j) ，空格位于 (k, l) 的情况下，到达目标点的最短路径。

建图直接看下一步往哪走。时间复杂度 $O(n^2 m^2 q)$ ，难以通过本题。

○○○

○ ○

○ ○

○ ○

【NOIP2013】华容道

所以直接设 $f(i, j, k)$ 表示指定棋子位于 (i, j) ，空格位于指定棋子的上下左右哪个方向。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○● ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

华容道

【NOIP2013】华容道

首先我们有一个暴力的想法。设 $f(i, j, k, l)$ 表示指定棋子位于 (i, j) ，空格位于 (k, l) 的情况下，到达目标点的最短路径。

建图直接看下一步往哪走。时间复杂度 $O(n^2 m^2 q)$ ，难以通过本题。

注意到，空格无论怎样，一定要移动到指定棋子周围才能起作用。

所以直接设 $f(i, j, k)$ 表示指定棋子位于 (i, j) ，空格位于指定棋子的上下左右哪个方向。

首先需要预处理状态之间的转移。跑一遍 bfs 即可。

总时间复杂度 $O(n^2 m^2 + nmq \log(nm))$ 。

○ ○

○ ○

○ ○



Royal Questions

CodeForces 875F

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	●○○	○○○	○○○○○
	○○	○○○	○○	

Royal Questions

CodeForces 875F

有 x 个王子 y 个公主，每一个公主喜欢其中两个王子，还有一个美丽度。

你需要将王子和公主配对，使得每一个公主都和自己喜欢的王子配对，并且配对的公主美丽度之和最大。

$$x, y \leq 200000$$

○○○



○ ○ ○ ○
○ ○ ○
○ ○ ○ ○ ○

Royal Questions

CodeForces 875F

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○●○	○○○	○○○○○
	○○	○○○	○○	

Royal Questions

CodeForces 875F

每一个公主可以抽象为一条边。

一个王子和一个公主配对，我们会发现答案形成了若干个基环树。（每一个连通块点数等于边数，刚好可以配对）

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○●○	○○○	○○○○○
	○○	○○○	○○	

Royal Questions

CodeForces 875F

每一个公主可以抽象为一条边。

一个王子和一个公主配对，我们会发现答案形成了若干个基环树。（每一个连通块点数等于边数，刚好可以配对）

实际上就是要求整张图的最大生成环套树。

○ ○

○ ○

○ ○



Royal Questions

CodeForces 875F

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○● ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Royal Questions

CodeForces 875F

先将所有边排序。接下来对于每一条边 (u, v) :

先将所有边排序。接下来对于每一条边 (u, v) :

- 若 u, v 不连通且分别位于两棵树内：加入这条边，得到一棵新的树。
- 若 u, v 不连通且分别位于一棵树和一棵基环树内：加入这条边，得到一个基环树。
- 若 u, v 不连通且分别位于两棵基环树内：无法加入。
- 若 u, v 连通且位于一棵树内：加入这条边，树变成基环树。
- 若 u, v 连通且位于一棵基环树内：无法加入。

CodeForces 875F

先将所有边排序。接下来对于每一条边 (u, v) :

- 若 u, v 不连通且分别位于两棵树内: 加入这条边, 得到一棵新的树。
- 若 u, v 不连通且分别位于一棵树和一棵基环树内: 加入这条边, 得到一个基环树。
- 若 u, v 不连通且分别位于两棵基环树内: 无法加入。
- 若 u, v 连通且位于一棵树内: 加入这条边, 树变成基环树。
- 若 u, v 连通且位于一棵基环树内: 无法加入。

只需要在并查集的基础上记录一个 `flag` 数组表示是否是基环树即可。

时间复杂度 $O(m \log m)$ 。

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
●○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

Mst

BZOJ 2238

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	●○○	○○	

Mst

BZOJ 2238

一张图， q 次询问删掉某条边之后最小生成树的大小。
询问相互独立。

$$n \leq 50000, m, q \leq 100000$$

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
●○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

Mst

BZOJ 2238

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Mst

BZOJ 2238

首先随便找一棵最小生成树。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○●○	○○	

Mst

BZOJ 2238

首先随便找一棵最小生成树。

如果删掉的边不在生成树上那么无影响。

删掉树边等价于求跨过这条边的非树边的最小权值。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○●○	○○	

Mst

BZOJ 2238

首先随便找一棵最小生成树。

如果删掉的边不在生成树上那么无影响。

删掉树边等价于求跨过这条边的非树边的最小权值。

如果用树剖的话，时间复杂度是两个 \log 的。

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○●

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
○○○○○

Mst

BZOJ 2238

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○●	○○	

Mst

BZOJ 2238

我们可以把其它所有的边按照边权从小到大排序。
接下来相当于每次要把未被标记的边打上标记。

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

●○○
○○○
○○○
○○○

Hard

○○○○
○○○
○○○○○

Cycle

HDU 5215

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	●○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Cycle

HDU 5215

有一个 n 个点 m 条边的无向图，问是否存在奇环/偶环。
 $n \leq 10^5, m \leq 3 \times 10^5$

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○●○
○○○
○○○
○○○

Hard

○○○○
○○○
○○○○○

Cycle

HDU 5215

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Cycle

HDU 5215

判断是否存在奇环很简单，只需要看是否是二分图即可。
如何判断是否存在偶环呢？

如何判断是否存在偶环呢?

我们首先建出来 dfs 树，然后对于每一条非树边 (u, v) (无向图，只能是返祖边):

- 若两点深度之差为奇数，那就找到了一个偶环。
- 若两个奇环相交，那么去掉相交的部分可以形成一个偶环。



○ ○

○ ○

○ ○

○ ○

H DU 5215

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○● ○○○ ○○○ ○○○	○○○○ ○○○ ○○○○○

Cycle

HDU 5215

我们直接把 u, v 这条路径上都打上标记，如果碰到了已经打过标记的点那就存在偶环。

暴力打标记即可，因为每一条边只会被访问一次。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○● ○○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Cycle

HDU 5215

我们直接把 u, v 这条路径上都打上标记，如果碰到了已经打过标记的点那就存在偶环。

暴力打标记即可，因为每一条边只会被访问一次。

时间复杂度 $O(n + m)$ 。



ICPC2018 青岛站 F

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ●○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Tournament

ICPC2018 青岛站 F

有 n 个人，要进行 k 轮决斗。

每一轮比赛你需要将这 n 个人两两配对，代表这两个人之间要进行一场决斗。

对于任意两个人，在所有 k 轮比赛中最多只能决斗一次。

若存在四个人 a, b, c, d ， a 和 b 在第 i 轮决斗， c 和 d 在第 i 轮决斗， a 和 c 在第 j 轮决斗，那么 b 和 d 必须也在第 j 轮决斗。

构造一组方案，无解输出 Impossible。

$1 \leq n, k \leq 1000$ ，多测， $\sum n, \sum k \leq 5000$

○ ○

○ ○

○ ○



ICPC2018 青岛站 F

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	●●○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Tournament

ICPC2018 青岛站 F

我们注意到如果 k 可行那么 $k - 1$ 也一定可行，所以相当于给你 n ，问最多能决斗多少轮。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ●○○ ○○○ ○○	○○○○ ○○○ ○○○○○

Tournament

ICPC2018 青岛站 F

我们注意到如果 k 可行那么 $k - 1$ 也一定可行，所以相当于给你 n ，问最多能决斗多少轮。

先来找一找规律：

- 当 n 是奇数的时候，一轮都不行。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○●○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Tournament

ICPC2018 青岛站 F

我们注意到如果 k 可行那么 $k-1$ 也一定可行，所以相当于给你 n ，问最多能决斗多少轮。

先来找一找规律：

- 当 n 是奇数的时候，一轮都不行。
- 当 n 是 2 的倍数但不是 4 的倍数的时候，可以进行第一轮，但是第二轮我们发现没法处理。（如果我们把第一轮和第二轮进行过决斗的都连边，那么相当于形成了若干个四元环）

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○● ○○○ ○○	○○○○ ○○○ ○○○○○

Tournament

ICPC2018 青岛站 F

我们注意到如果 k 可行那么 $k-1$ 也一定可行，所以相当于给你 n ，问最多能决斗多少轮。

先来找一找规律：

- 当 n 是奇数的时候，一轮都不行。
- 当 n 是 2 的倍数但不是 4 的倍数的时候，可以进行第一轮，但是第二轮我们发现没法处理。（如果我们把第一轮和第二轮进行过决斗的都连边，那么相当于形成了若干个四元环）
- 当 n 是 4 的倍数的时候我们至少能构造出三轮：1-2 3-4, 1-3 2-4 1-4 2-3，但如果不是 8 的倍数那就构造不出第四轮。（第四轮一定会把连通块两两合并）

○○○



○ ○ ○ ○
○ ○ ○
○ ○ ○ ○ ○

ICPC2018 青岛站 F

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○●	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Tournament

ICPC2018 青岛站 F

当 n 是 2^k 的倍数时，我们至少能构造出 $2^k - 1$ 轮。只需要每一个大小为 2^k 的连通块分别构造即可。

所以只需要看 n 的二进制末尾有几个 0，就能立刻得到答案。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○● ○○○ ○○	○○○○ ○○○ ○○○○○

Tournament

ICPC2018 青岛站 F

当 n 是 2^k 的倍数时，我们至少能构造出 $2^k - 1$ 轮。只需要每一个大小为 2^k 的连通块分别构造即可。

所以只需要看 n 的二进制末尾有几个 0，就能立刻得到答案。

输出方案上，教给大家一个小技巧：把 2^k 个人分别标号为 0 到 $2^k - 1$ ，每一轮我们枚举一个 $[1, 2^k - 1]$ 之间的数 v ，然后选取所有二进制异或和为 v 的数进行决斗。

时间复杂度 $O(nk)$ 。

○○○



○ ○ ○ ○
○ ○ ○
○ ○ ○ ○ ○

CodeForces 1335F

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ●○○ ○○	○○○○ ○○○ ○○○○○

Robots on a Grid

CodeForces 1335F

有一个 $n \times m$ 的网格，每一个格子有一个颜色（黑白）和一个方向（上下左右）。

你需要在若干个格子上放置机器人。接下来的每一个时刻，所有机器人会同时沿着格子上的方向移动到相邻的格子。数据保证机器人不会移出格子。

你需要保证在任意时刻，任意一个格子上最多只能有一个机器人。

你需要使得放置的机器人数量尽可能多。在此基础上，你还需要使得黑色格子上的机器人数量尽可能多。

$$nm \leq 10^6$$



A 4x3 grid of circles. The third row, middle circle is filled black.

Robots on a Grid

CodeForces 1335F

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ●○○ ○○	○○○○ ○○○ ○○○○○

Robots on a Grid

CodeForces 1335F

每一个点有一个后继，先连边。

连完之后我们可以发现，每一个连通块都是一个环上挂着若干棵树，树上的边都是由子节点指向父节点。（其实就是基环树）。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○●	○○○○○
	○○	○○○	○○	

Robots on a Grid

CodeForces 1335F

每一个点有一个后继，先连边。

连完之后我们可以发现，每一个连通块都是一个环上挂着若干棵树，树上的边都是由子节点指向父节点。(其实就是基环树)。

随着时间的流逝最后所有机器人都会走到环上，也就是说对于每一个连通块，我们最多只能放环长那么多机器人。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○●	○○○○○
	○○	○○○	○○	

Robots on a Grid

CodeForces 1335F

每一个点有一个后继，先连边。

连完之后我们可以发现，每一个连通块都是一个环上挂着若干棵树，树上的边都是由子节点指向父节点。(其实就是基环树)。

随着时间的流逝最后所有机器人都会走到环上，也就是说对于每一个连通块，我们最多只能放环长那么多机器人。

而为了让黑色格子上机器人数量尽可能多，我们可能还需要做一些处理。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○● ○○	○○○○ ○○○ ○○○○○

Robots on a Grid

CodeForces 1335F

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○●
○○○

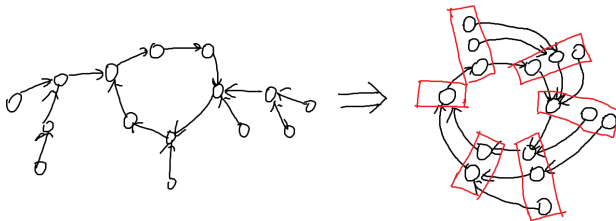
Hard

○○○○
○○○
○○○○○

Robots on a Grid

CodeForces 1335F

尝试了好几种表达，怎么都说不清楚，直接上图吧：



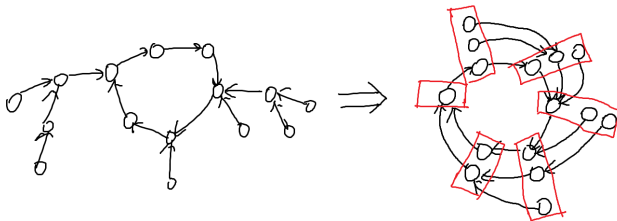
红框框起来的点里只能选择一个放置机器人。我们只需要看每一个红框里是否有黑色点即可。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○●	○○○○○
	○○	○○○	○○	

Robots on a Grid

CodeForces 1335F

尝试了好几种表达，怎么都说不清楚，直接上图吧：



红框框起来的点里只能选择一个放置机器人。我们只需要看每一个红框里是否有黑色点即可。

至于代码实现，我们可以随便断掉环上的一条边，当成一棵树进行 dfs，最后把深度全都对一个值取模。

○ ○

○ ○

○ ○



Quantifier Question

CodeForces 1344C

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ●○	○○○○ ○○○ ○○○○○

Quantifier Question

CodeForces 1344C

有 n 个数 x_1, x_2, \dots, x_n , 还有 m 个有序数对 (u_i, v_i) 。

你需要在每一个数 x_i 前面添加一个符号 Q_i , 其中 Q_i 只能是 \forall (任意) 或者 \exists (存在)。

你需要让下面这个式子成立:

$$Q_1 x_1, Q_2 x_2, \dots, Q_n x_n, x_{u_1} < x_{v_1}, x_{u_2} < x_{v_2}, \dots, x_{u_m} < x_{v_m}$$

在此基础上, 你需要让 \forall (任意) 的数量尽可能多。

输出一组解, 无解输出 -1 。

$$2 \leq n \leq 2 \times 10^5, 1 \leq m \leq 2 \times 10^5$$



CodeForces 1344C

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○●	

Quantifier Question

CodeForces 1344C

这道题难度几乎都在读题上 ==
建图，则如果出现环直接不可能。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	●○	

Quantifier Question

CodeForces 1344C

这道题难度几乎都在读题上 ==

建图，则如果出现环直接不可能。

对于一个偏序关系 $x_{k_1} < x_{k_2} < \dots < x_{k_m}$ ，只能有一个 \forall ，并且只可能位于最开始。

也就是说对于图上的一条链，只有其中编号最小的点可以是 \forall ，其余都得是 \exists 。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ●●	○○○○ ○○○ ○○○○○

Quantifier Question

CodeForces 1344C

这道题难度几乎都在读题上 ==

建图，则如果出现环直接不可能。

对于一个偏序关系 $x_{k_1} < x_{k_2} < \dots < x_{k_m}$ ，只能有一个 \forall ，并且只可能位于最开始。

也就是说对于图上的一条链，只有其中编号最小的点可以是 \forall ，其余都得是 \exists 。

换句话说，对于一个点，它可以是 \forall ，当且仅当所有可以到达它的点以及所有它可以到达的点，编号都大于它。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	●	

Quantifier Question

CodeForces 1344C

这道题难度几乎都在读题上 ==

建图，则如果出现环直接不可能。

对于一个偏序关系 $x_{k_1} < x_{k_2} < \dots < x_{k_m}$ ，只能有一个 \forall ，并且只可能位于最开始。

也就是说对于图上的一条链，只有其中编号最小的点可以是 \forall ，其余都得是 \exists 。

换句话说，对于一个点，它可以是 \forall ，当且仅当所有可以到达它的点以及所有它可以到达的点，编号都大于它。

正反建两遍图，对于每一个点 DP 出能够到达它的点的最小值，如果这个最小值大于它那它就是 \forall ，否则就是 \exists 。

时间复杂度 $O(n + m)$ 。



CodeForces Gym 100269K

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	●○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Kids in a Friendly Class

CodeForces Gym 100269K

一张图有黑点和白点，每个黑点有 a 条边和黑点相连 b 条边和白点相连，每个白点有 c 条边和黑点相连有 d 条边和白点相连。

求一个方案使总点数最少。

$$a, b, c, d \leq 50$$



CodeForces Gym 100269K

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○●○○
○○○
○○○○○

Kids in a Friendly Class

CodeForces Gym 100269K

一看数据范围那么小，干脆先枚举有多少个黑点。

如果黑点数量为 n ，那么白点肯定要有 $\frac{nb}{c}$ 个。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○●○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Kids in a Friendly Class

Codeforces Gym 100269K

一看数据范围那么小，干脆先枚举有多少个黑点。

如果黑点数量为 n ，那么白点肯定要有 $\frac{nb}{c}$ 个。

黑白点之间的边需要用网络流来解决，我们就不讲了。

怎么构造同色点之间的边呢？

○ ○

○ ○

○ ○

○ ○

○ ○

○ ○

○ ○

CodeForces Gym 100269K

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○●○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Kids in a Friendly Class

CodeForces Gym 100269K

问题转化为一张 n 个点的图，告诉你每一个点的度数求方案。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○●○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Kids in a Friendly Class

CodeForces Gym 100269K

问题转化为一张 n 个点的图，告诉你每一个点的度数求方案。

错误贪心：每次取两个剩余度数最大的点相连。

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○●○
○○○
○○○○○

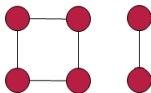
Kids in a Friendly Class

CodeForces Gym 100269K

问题转化为一张 n 个点的图，告诉你每一个点的度数求方案。

错误贪心：每次取两个剩余度数最大的点相连。

Hack:





CodeForces Gym 100269K

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○●
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Kids in a Friendly Class

CodeForces Gym 100269K

正确的思路：每次找度数最大的点，假设度数为 d ，那就找到剩下的度数最大的 d 个点，然后连边并且删掉这个点。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○●
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○○○
	○○	○○○	○○	

Kids in a Friendly Class

CodeForces Gym 100269K

正确的思路：每次找度数最大的点，假设度数为 d ，那就找到剩下的度数最大的 d 个点，然后连边并且删掉这个点。

这个算法也可以用来构造任意给定度数的图。

○ ○

○ ○

○ ○



Makoto Soejima's Contest 4

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○○	○○○○ ●○○ ○○○○○

Jump

Makoto Soejima's Contest 4

数轴上有 n 个跳跃中心 a_1, a_2, \dots, a_n , 你可以一步从 x 跳到 $2a_i - x$ 。

有 q 次询问, 每次问你从 S 到 T 最少跳几次。无解输出 -1 。

$$1 \leq n \leq 200, 0 \leq a_i, S, T \leq 10000, 1 \leq q \leq 100000$$

○○○



Makoto Soejima's Contest 4

找规律！

假设我们依次跳过了 a_{k_1}, a_{k_2}, \dots :

- 第一次跳完变成了 $2a_{k_1} - S$;
- 第二次跳完变成了 $2a_{k_2} - 2a_{k_1} + S$;
- 第三次跳完变成了 $2a_{k_3} - 2a_{k_2} + 2a_{k_1} - S$;
-

○○○



○○○
○○
○○○
○○○

○ ○ ○
○ ○ ○
○ ○ ○
○ ○

Makoto Soejima's Contest 4

Makoto Soejima's Contest 4

最后到达 T 只有两种情况：

- 1 $2a_{k_m} - 2a_{k_{m-1}} + \cdots + 2a_{k_1} - S = T$
- 2 $2a_{k_m} - 2a_{k_{m-1}} + \cdots - 2a_{k_1} + S = T$

Makoto Soejima's Contest 4

最后到达 T 只有两种情况：

- 1 $2a_{k_m} - 2a_{k_{m-1}} + \cdots + 2a_{k_1} - S = T$
- 2 $2a_{k_m} - 2a_{k_{m-1}} + \cdots - 2a_{k_1} + S = T$

也就是说

- 1 $a_{k_m} - a_{k_{m-1}} + \cdots + a_{k_1} = \frac{T+S}{2}$
- 2 $a_{k_m} - a_{k_{m-1}} + \cdots - a_{k_1} = \frac{T-S}{2}$

Makoto Soejima's Contest 4

最后到达 T 只有两种情况：

- 1 $2a_{k_m} - 2a_{k_{m-1}} + \cdots + 2a_{k_1} - S = T$
- 2 $2a_{k_m} - 2a_{k_{m-1}} + \cdots - 2a_{k_1} + S = T$

也就是说

- 1 $a_{k_m} - a_{k_{m-1}} + \cdots + a_{k_1} = \frac{T+S}{2}$
- 2 $a_{k_m} - a_{k_{m-1}} + \cdots - a_{k_1} = \frac{T-S}{2}$

也就是从 0 开始到 $\frac{T+S}{2}$ 和 $\frac{T-S}{2}$ 。

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○● ○○○○○
Jump				

Makoto Soejima's Contest 4

最后到达 T 只有两种情况：

- 1 $2a_{k_m} - 2a_{k_{m-1}} + \cdots + 2a_{k_1} - S = T$
- 2 $2a_{k_m} - 2a_{k_{m-1}} + \cdots - 2a_{k_1} + S = T$

也就是说

- 1 $a_{k_m} - a_{k_{m-1}} + \cdots + a_{k_1} = \frac{T+S}{2}$
- 2 $a_{k_m} - a_{k_{m-1}} + \cdots - a_{k_1} = \frac{T-S}{2}$

也就是从 0 开始到 $\frac{T+S}{2}$ 和 $\frac{T-S}{2}$ 。

从 0 出发进行 bfs，记录一下到每一个点走奇数步的距离和走偶数步的距离。

时间复杂度 $O(10000n + q)$ 。

○ ○

○ ○

○ ○



Nastya and Time Machine

CodeForces 1340D

Trivial

○○
○○
○○

Easy

○○
○○
○○
○○

Medium

○○○
○○
○○○
○○○

Not Hard

○○○
○○○
○○○
○○

Hard

○○○○
○○○
●○○○

Nastya and Time Machine

CodeForces 1340D

有一棵 n 个点的树，你从 1 号点开始遍历，最后回到 1 号点。通过一条边需要的时间是 1，初始时间是 0。

你有一台时间机器，可以让你在不改变所在位置的情况下穿越到任意一个小于当前时间且大于等于 0 的时间点。（必须是整数）

由于不能违背宇宙法则，你不能在两次在同一个时间点位于同一个节点。

你需要设置一个遍历次序，满足所有时间点的最大值最小。

$$1 \leq n \leq 10^5$$

○ ○

○ ○

○ ○



Nastya and Time Machine

CodeForces 1340D

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○●○○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

首先考虑一些特殊情况。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○●○○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

首先考虑一些特殊情况。
一条链?

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○●○○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

首先考虑一些特殊情况。

一条链?

除了只有两个点的答案为 1，其余答案都是 2。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○●○○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

首先考虑一些特殊情况。

一条链？

除了只有两个点的答案为 1，其余答案都是 2。

一个菊花图？

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○●○○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

首先考虑一些特殊情况。

一条链？

除了只有两个点的答案为 1，其余答案都是 2。

一个菊花图？

答案即为 $n - 1$ 。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○●○○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

首先考虑一些特殊情况。

一条链？

除了只有两个点的答案为 1，其余答案都是 2。

一个菊花图？

答案即为 $n - 1$ 。

似乎节点的度数会对答案造成很大的影响。

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○●○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

Trivial	Easy	Medium	Not Hard	Hard
○○ ○○ ○○	○○ ○○ ○○ ○○	○○○ ○○ ○○○ ○○○	○○○ ○○○ ○○○ ○○	○○○○ ○○○ ○○○ ○○●○○

Nastya and Time Machine

CodeForces 1340D

设最大时间为 T 。

对于一棵子树，如果你在时间点 $t + 1$ 到达这棵子树的根，那么一定可以在时间点 t 遍历完整棵子树并且回到根。（在根上用一次穿越）

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○●○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

设最大时间为 T 。

对于一棵子树，如果你在时间点 $t + 1$ 到达这棵子树的根，那么一定可以在时间点 t 遍历完整棵子树并且回到根。（在根上用一次穿越）

也就是说对于一个点，一个分支只会消耗 1 单位时间。（ t 出去， $t + 1$ 回来）

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○●○○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

设最大时间为 T 。

对于一棵子树，如果你在时间点 $t+1$ 到达这棵子树的根，那么一定可以在时间点 t 遍历完整棵子树并且回到根。（在根上用一次穿越）

也就是说对于一个点，一个分支只会消耗 1 单位时间。（ t 出去， $t+1$ 回来）

唯一的特殊情况是 $t = T$ 的情况，这时候我们没法直接进入子树，只能先穿越到 0 时间，然后在 1 时间回来。

进一步发现其实 T 和 0 没区别，因为无论如何下一步我们一定要穿越到 0。（除非已经到终点了，这时候我们可以穿越到 0 使得整个路径成为一个环）

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○●○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

Trivial	Easy	Medium	Not Hard	Hard
○○	○○	○○○	○○○	○○○○
○○	○○	○○	○○○	○○○
○○	○○	○○○	○○○	○○○●○
	○○	○○○	○○	

Nastya and Time Machine

CodeForces 1340D

我们发现，对于一个度数为 d 的点，我们只需要 d 的时刻就够了！

CodeForces 1340D

我们发现，对于一个度数为 d 的点，我们只需要 d 的时刻就够了！

- 1 t 时刻进入并且往第一个分支走， $t+1$ 回来；
- 2 $t+1$ 时刻往第二个分支走， $t+2$ 回来；
- 3
- 4 $d-1$ 时刻往第 $d-t$ 个分支走， d 回来；
- 5 传送到 0 时刻，往第 $d-t+1$ 个分支走，1 回来；
- 6
- 7 $t-2$ 时刻往第 $d-1$ 个分支走， $t-1$ 回来。
- 8 $t-1$ 时刻遍历完整棵子树并且回到根。

○ ○

○ ○

○ ○



○ ○ ○ ○
○ ○ ○
○ ○ ○ ○ ●

Nastya and Time Machine

CodeForces 1340D

CodeForces 1340D

答案即为 $\max d_i$ 。顺着这个思路构造就行了。每一个点设一个 cur 表示当前到了第几个时刻。

时间复杂度 $O(n)$ 。