

2013 年 NOIP 试题分析

ljfcnyali

August 16, 2019

Day1 T1

Day1 T1 转圈游戏

有 n 个小伙伴，编号为 $0 \rightarrow n-1$ 围成一个环，每轮每个小伙伴移动 m 个位置，即第 0 个移动到第 m 个位置，第 $n-m$ 移动到第 0 个位置，一共进行了 10^k 轮，求编号为 x 的小伙伴的位置

30' 满足 $k \leq 7$

80' 满足 $k \leq 10^7$

100' 满足 $n \leq 10^7, k \leq 10^9$

Day1 T1 转圈游戏

30' 暴力移动每一个人即可

Day1 T1 转圈游戏

30' 暴力移动每一个人即可

80' 考虑编号为 x 的伙伴，第一轮会移动到 $x + m(\bmod n)$ ，第二次移动到 $x + 2m(\bmod n)$ ，则第 10^k 移动到 $x + 10^k m(\bmod n)$

Day1 T1 转圈游戏

30' 暴力移动每一个人即可

80' 考虑编号为 x 的伙伴，第一轮会移动到 $x + m(\bmod n)$ ，第二次移动到 $x + 2m(\bmod n)$ ，则第 10^k 移动到 $x + 10^k m(\bmod n)$

所以直接 $O(k)$ 的求出 10^k

Day1 T1 转圈游戏

30' 暴力移动每一个人即可

80' 考虑编号为 x 的伙伴，第一轮会移动到 $x + m(\bmod n)$ ，第二次移动到 $x + 2m(\bmod n)$ ，则第 10^k 移动到 $x + 10^k m(\bmod n)$

所以直接 $O(k)$ 的求出 10^k

100' 发现 k 较大，问题即转化为如何快速求 $10^k(\bmod n)$ ，快速幂套模板即可

Day1 T1 转圈游戏

总结一下这道题做法：将式子推出来，用快速幂优化时间复杂度

Day1 T1 转圈游戏

总结一下这道题做法：将式子推出来，用快速幂优化时间复杂度

另外这道题可以将循环节推出来，周期长度为 $n/\gcd(n,m)$ ，然后随便怎么做都可以

Day1 T1 转圈游戏

总结一下这道题做法：将式子推出来，用快速幂优化时间复杂度

另外这道题可以将循环节推出来，周期长度为 $n/\gcd(n,m)$ ，然后随便怎么做都可以

Day T1 还是相信大多数人可以在半个小时内解决掉的 ...

Day1 T2

Day1 T2 火柴排队

有两盒每盒 n 根火柴，每根火柴有一个高度，将每盒火柴排成一列，同一列火柴的高度互不相同，定义两列火柴的距离为 $\sum_{i=1}^n (a_i - b_i)^2$

Day1 T2 火柴排队

有两盒每盒 n 根火柴，每根火柴有一个高度，将每盒火柴排成一列，同一列火柴的高度互不相同，定义两列火柴的距离为 $\sum_{i=1}^n (a_i - b_i)^2$ 每列火柴中相邻两根火柴可以交换，求出使得这两列火柴距离最小的交换次数，对 99999997 取模

Day1 T2 火柴排队

有两盒每盒 n 根火柴，每根火柴有一个高度，将每盒火柴排成一列，同一列火柴的高度互不相同，定义两列火柴的距离为 $\sum_{i=1}^n (a_i - b_i)^2$

每列火柴中相邻两根火柴可以交换，求出使得这两列火柴距离最小的交换次数，对 99999997 取模

60' 满足 $n \leq 10^3$

Day1 T2 火柴排队

有两盒每盒 n 根火柴，每根火柴有一个高度，将每盒火柴排成一列，同一列火柴的高度互不相同，定义两列火柴的距离为 $\sum_{i=1}^n (a_i - b_i)^2$

每列火柴中相邻两根火柴可以交换，求出使得这两列火柴距离最小的交换次数，对 99999997 取模

60' 满足 $n \leq 10^3$

100' 满足 $n \leq 10^6$

Day1 T2 火柴排队

第一个性质：只需要交换第一列

Day1 T2 火柴排队

第一个性质：只需要交换第一列

第二个性质：假设第二列中第 i 根火柴的高度排名为 rank_i ，当且仅当第一列中第 i 根火柴的高度排名也为 rank_i 时两列火柴距离最短

Day1 T2 火柴排队

第一个性质：只需要交换第一列

第二个性质：假设第二列中第 i 根火柴的高度排名为 rank_i ，当且仅当第一列中第 i 根火柴的高度排名也为 rank_i 时两列火柴距离最短

证明：先拆一下式子，发现影响答案的是 $2 \times \sum_{i=1}^n a_i b_i$ ，我们需要最大化这个值来使得答案最小

Day1 T2 火柴排队

第一个性质：只需要交换第一列

第二个性质：假设第二列中第 i 根火柴的高度排名为 rank_i ，当且仅当第一列中第 i 根火柴的高度排名也为 rank_i 时两列火柴距离最短

证明：先拆一下式子，发现影响答案的是 $2 \times \sum_{i=1}^n a_i b_i$ ，我们需要最大化这个值来使得答案最小

假设 $a_i < a_j, b_i < b_j$ ，可以得到 $b_i(a_j - a_i) < b_j(a_j - a_i)$

Day1 T2 火柴排队

第一个性质：只需要交换第一列

第二个性质：假设第二列中第 i 根火柴的高度排名为 rank_i ，当且仅当第一列中第 i 根火柴的高度排名也为 rank_i 时两列火柴距离最短

证明：先拆一下式子，发现影响答案的是 $2 \times \sum_{i=1}^n a_i b_i$ ，我们需要最大化这个值来使得答案最小

假设 $a_i < a_j, b_i < b_j$ ，可以得到 $b_i(a_j - a_i) < b_j(a_j - a_i)$

移下项即得 $a_i b_i + a_j b_j > a_i b_j + a_j b_i$

Day1 T2 火柴排队

现在这个问题就很好解决了，首先将 a, b 都离散化，因为答案只关心 a_i, b_i 的位置关系

Day1 T2 火柴排队

现在这个问题就很好解决了，首先将 a, b 都离散化，因为答案只关心 a_i, b_i 的位置关系

令 $p[i]$ 等于 a_i 与之值相等的 b_j 的位置，只需求出使得 p 序列升序的最小交换次数

Day1 T2 火柴排队

现在这个问题就很好解决了，首先将 a, b 都离散化，因为答案只关心 a_i, b_i 的位置关系

令 $p[i]$ 等于 a_i 与之值相等的 b_j 的位置，只需求出使得 p 序列升序的最小交换次数

答案即 p 数组的逆序对，树状数组和归并排序都可以求出答案

Day1 T2 火柴排队

这道题关键在于猜结论并进行证明，这个结论其实非常套路

Day1 T2 火柴排队

这道题关键在于猜结论并进行证明，这个结论其实非常套路

后面的处理还是需要一点技巧，离散化 + 逆序对也是逆序对题目的经典套路了

Day1 T2 火柴排队

这道题关键在于猜结论并进行证明，这个结论其实非常套路

后面的处理还是需要一点技巧，离散化 + 逆序对也是逆序对题目的经典套路了

这题算结论题么（类似小凯的疑惑）???

Day1 T3

Day1 T3 货车运输

有 n 座城市 m 条道路，每条道路有限重 w_i ，现在有 q 辆货车运送货物，司机想知道每台车在不超过限重的情况下最多可以运送多重的货物

Day1 T3 货车运输

有 n 座城市 m 条道路，每条道路有限重 w_i ，现在有 q 辆货车运送货物，司机想知道每台车在不超过限重的情况下最多可以运送多重的货物

100' 满足 $n \leq 10^4, m \leq 5 \times 10^4, q \leq 3 \times 10^4$

方法一

方法一

观察可发现一个性质：经过的路径一定是最大生成树上的一个子路径

方法一

观察可发现一个性质：经过的路径一定是最大生成树上的一个子路径

先将每个连通块分开，我们只需要考虑在一个连通块中的情况

方法一

观察可发现一个性质：经过的路径一定是最大生成树上的一个子路径

先将每个连通块分开，我们只需要考虑在一个连通块中的情况

对于每一个连通块，求出最大生成树，并且将 LCA 进行预处理，维护每个节点的祖先和最小边权，每个询问的答案一定是 $u \rightarrow v$ 在最大生成树的路径上的边权最小值

方法一

观察可发现一个性质：经过的路径一定是最大生成树上的一个子路径

先将每个连通块分开，我们只需要考虑在一个连通块中的情况

对于每一个连通块，求出最大生成树，并且将 LCA 进行预处理，维护每个节点的祖先和最小边权，每个询问的答案一定是 $u \rightarrow v$ 在最大生成树的路径上的边权最小值

为什么我觉得这道题又是一道结论题呢???

方法二

方法二

Kruskal 重构树一般解决任意两点上边权的最大最小值

方法二

Kruskal 重构树一般解决任意两点上边权的最大最小值

在做 Kruskal 算法的时候，每次连接两个不在同一连通块的点，就将这两个点的根与一个新建的节点各连上一条边，新建的节点权值等于这条边的权值

方法二

Kruskal 重构树一般解决任意两点上边权的最大最小值

在做 Kruskal 算法的时候，每次连接两个不在同一连通块的点，就将这两个点的根与一个新建的节点各连上一条边，新建的节点权值等于这条边的权值

这样建出一棵树就有很多方法解决问题了

方法三

方法三

可持久化并查集...

方法三

可持久化并查集...

方便维护的话套个 `set`

Day2 T1

Day2 T1 积木大赛

需要搭建一座宽度为 n 的大厦，大厦可以看做由 n 块宽度为 1 的积木组成，第 i 块积木最终高度需要 h_i ，每次可以将任意 $[l, r]$ 区间的积木高度加 1 ，求从所有为积木为空到目标状态最小需要几次操作

Day2 T1 积木大赛

需要搭建一座宽度为 n 的大厦，大厦可以看做由 n 块宽度为 1 的积木组成，第 i 块积木最终高度需要 h_i ，每次可以将任意 $[l, r]$ 区间的积木高度加 1 ，求从所有为积木为空到目标状态最小需要几次操作

100' 满足 $n \leq 10^5$

Day2 T1 积木大赛

这道题不要多想，Day2 T1 不可能很难的，数据结构、cdq 分治都不需要的（虽然还是可以做），直接想贪心 & 差分 & 模拟就好了

Day2 T1 积木大赛

这道题不要多想，Day2 T1 不可能很难的，数据结构、cdq 分治都不需要的（虽然还是可以做），直接想贪心 & 差分 & 模拟就好了
我一开始想到的是差分，答案其实就是差分数组中的正值之和

Day2 T1 积木大赛

这道题不要多想，Day2 T1 不可能很难的，数据结构、cdq 分治都不需要的（虽然还是可以做），直接想贪心 & 差分 & 模拟就好了

我一开始想到的是差分，答案其实就是差分数组中的正值之和

感性理解一下，每次可以将差分数组中一个数 $+1$ ，一个数 -1 ，那么每个正数都是要进行操作的

Day2 T1 积木大赛

这道题不要多想，Day2 T1 不可能很难的，数据结构、cdq 分治都不需要的（虽然还是可以做），直接想贪心 & 差分 & 模拟就好了

我一开始想到的是差分，答案其实就是差分数组中的正值之和

感性理解一下，每次可以将差分数组中一个数 $+1$ ，一个数 -1 ，那么每个正数都是要进行操作的

这个差分方法可以拓展一下，发现按照差分思路只有 $h_i > h_{i-1}$ 的时候有贡献，加上就可以了

用贪心的思路思考也可以

Day2 T1 积木大赛

用贪心的思路思考也可以

假设当前考虑到第 i 个位置，前 $1 \rightarrow i-1$ 都已经摆放完毕，有两种情况

用贪心的思路思考也可以

假设当前考虑到第 i 个位置，前 $1 \rightarrow i-1$ 都已经摆放完毕，有两种情况

第一种： $h_{i-1} \geq h_i$ 时如果摆放好了 $i-1$ 位置，那么 i 位置一定可以摆好

用贪心的思路思考也可以

假设当前考虑到第 i 个位置，前 $1 \rightarrow i-1$ 都已经摆放完毕，有两种情况

第一种： $h_{i-1} \geq h_i$ 时如果摆放好了 $i-1$ 位置，那么 i 位置一定可以摆好

第二种： $h_{i-1} < h_i$ 时答案显然需要加上 $h_i - h_{i-1}$

思考一下模拟

思考一下模拟

假设当前考虑区间 $[l, r]$ ，我们将这个区间减去区间最小值，然后把每个变为 0 的分开继续模拟

思考一下模拟

假设当前考虑区间 $[l, r]$ ，我们将这个区间减去区间最小值，然后把每个变为 0 的分开继续模拟

为了保证复杂度正确，区间最小值用线段树维护一下

思考一下模拟

假设当前考虑区间 $[l, r]$ ，我们将这个区间减去区间最小值，然后把每个变为 0 的分开继续模拟

为了保证复杂度正确，区间最小值用线段树维护一下

也可以用并查集来模拟

Day2 T2

有 n 株花，花的高度为 h_i ，移走一部分花后剩下的花高度依次为 $g_1, g_2 \dots g_m$ ，希望移走最少的花，满足 $g_{2i} > g_{2i-1}, g_{2i} > g_{2i+1} >$ 或 $g_{2i} < g_{2i-1}, g_{2i} < g_{2i+1}$

有 n 株花，花的高度为 h_i ，移走一部分花后剩下的花高度依次为 $g_1, g_2 \dots g_m$ ，希望移走最少的花，满足 $g_{2i} > g_{2i-1}, g_{2i} > g_{2i+1} >$ 或 $g_{2i} < g_{2i-1}, g_{2i} < g_{2i+1}$
100' 满足 $n \leq 10^5$

终于遇见了一道 DP 题，令 $dp[i][0]$ 表示 i 为波峰的最大答案， $dp[i][1]$ 表示 i 为波谷的最大答案

终于遇见了一道 DP 题，令 $dp[i][0]$ 表示 i 为波峰的最大答案， $dp[i][1]$ 表示 i 为波谷的最大答案

转移方程很显然： $dp[i][0] = \max_{a[j] < a[i]} (dp[j][1] + 1)$ ，
 $dp[i][1] = \max_{a[j] > a[i]} (dp[j][0] + 1)$

终于遇见了一道 DP 题，令 $dp[i][0]$ 表示 i 为波峰的最大答案， $dp[i][1]$ 表示 i 为波谷的最大答案

转移方程很显然： $dp[i][0] = \max_{a[j] < a[i]} (dp[j][1] + 1)$ ，
 $dp[i][1] = \max_{a[j] > a[i]} (dp[j][0] + 1)$

权值线段树维护一下区间最值就可以 $O(\log_2 n)$ 转移了，注意要离散化或动态开点，不然会爆空间

终于遇见了一道 DP 题，令 $dp[i][0]$ 表示 i 为波峰的最大答案， $dp[i][1]$ 表示 i 为波谷的最大答案

转移方程很显然： $dp[i][0] = \max_{a[j] < a[i]} (dp[j][1] + 1)$ ，
 $dp[i][1] = \max_{a[j] > a[i]} (dp[j][0] + 1)$

权值线段树维护一下区间最值就可以 $O(\log_2 n)$ 转移了，注意要离散化或动态开点，不然会爆空间

但我总觉得 $O(n)$ 的 DP 和贪心有点迷，不过 $O(n \log_2 n)$ 的 DP 还是很好想的

Day2 T3

有一个 $n \times m$ 的棋盘，有且只有一个格子是空的，其余 $n \times m - 1$ 个格子上都有一个 1×1 的棋子，其中有一部分棋子是固定的，而另一部分棋子可以移动，求将一个棋子移动到目标位置的最小步数，有多组询问

有一个 $n \times m$ 的棋盘，有且只有一个格子是空的，其余 $n \times m - 1$ 个格子上都有一个 1×1 的棋子，其中有一部分棋子是固定的，而另一部分棋子可以移动，求将一个棋子移动到目标位置的最小步数，有多组询问

100' 满足 $1 \leq n, m \leq 30, q \leq 500$

好吧 A^* 是可以过的，估价函数可以写成目标棋子到目标位置的距离

好吧 A^* 是可以过的，估价函数可以写成目标棋子到目标位置的距离
这种题的一种套路是先将空白格子移动到目标棋子的位置，然后两个
捆绑着移动

好吧 A^* 是可以过的，估价函数可以写成目标棋子到目标位置的距离
这种题的一种套路是先将空白格子移动到目标棋子的位置，然后两个
捆绑着移动

将空白格子移动到目标棋子的 4 个方向可以直接 BFS 找出，这部分
比较简单

好吧 A^* 是可以过的，估价函数可以写成目标棋子到目标位置的距离
这种题的一种套路是先将空白格子移动到目标棋子的位置，然后两个
捆绑着移动

将空白格子移动到目标棋子的 4 个方向可以直接 BFS 找出，这部分
比较简单

设 $dis[i][j][k]$ 表示目标棋子在 (i,j) ，空白格子在 (i,j) 的 k 方
向的最少移动次数 (k 表示 4 个方向)

好吧 A^* 是可以过的，估价函数可以写成目标棋子到目标位置的距离
这种题的一种套路是先将空白格子移动到目标棋子的位置，然后两个
捆绑着移动

将空白格子移动到目标棋子的 4 个方向可以直接 BFS 找出，这部分
比较简单

设 $dis[i][j][k]$ 表示目标棋子在 (i,j) ，空白格子在 (i,j) 的 k 方
向的最少移动次数 (k 表示 4 个方向)

因为棋盘是不会变化的，所以可以预处理出所有从 i,j,k 可以移动
到的状态，对每一个询问跑 Dijkstra 就可以了

试题分析

2013 年考察的知识点有：数论（快速幂）；树状数组或归并排序求逆序对；最大生成树；LCA 和倍增预处理；差分；贪心；线段树优化 DP；搜索优化和最短路

2013 年考察的知识点有：数论（快速幂）；树状数组或归并排序求逆序对；最大生成树；LCA 和倍增预处理；差分；贪心；线段树优化 DP；搜索优化和最短路

这一年考察的还是很全面的，信息竞赛经常使用到的算法都有所涉猎，思维难度和代码难度适中，是近几年来不可多得优秀试题（我说的就是 18 年）

试题分析

2013 年考察的知识点有：数论（快速幂）；树状数组或归并排序求逆序对；最大生成树；LCA 和倍增预处理；差分；贪心；线段树优化 DP；搜索优化和最短路

这一年考察的还是很全面的，信息竞赛经常使用到的算法都有所涉猎，思维难度和代码难度适中，是近几年来不可多得优秀试题（我说的就是 18 年）

暴力分人均 360 分

Day1 出题方向主要是对结论的考察，只要想到这道题的结论就可以快速的想清整个题目的思路并完成代码编写

Day1 出题方向主要是对结论的考察，只要想到这道题的结论就可以快速的想清整个题目的思路并完成代码编写

所以相信：人有多大胆，地有多大产!!!

Day1 出题方向主要是对结论的考察，只要想到这道题的结论就可以快速的想清整个题目的思路并完成代码编写

所以相信：人有多大胆，地有多大产!!!

Day2 反而让人很容易忽略简洁的写法，T1 的贪心其实很明显却很容易忽略，T2 的贪心写法考场上有人敢写么？T3 的思路实在是太巧妙很难想到，大部分人还是会选择搜索不会深思吧

Day1 出题方向主要是对结论的考察，只要想到这道题的结论就可以快速的想清整个题目的思路并完成代码编写

所以相信：人有多大胆，地有多大产!!!

Day2 反而让人很容易忽略简洁的写法，T1 的贪心其实很明显却很容易忽略，T2 的贪心写法考场上有人敢写么？T3 的思路实在是太巧妙很难想到，大部分人还是会选择搜索不会深思吧

而且这套试题的套路分布非常密集，如果富有经验的选手应该可以快速的想明白各个题目考察的知识点

讲一下各个题目易错点：

试题分析

讲一下各个题目易错点：

Day1 T1 注意下标从 0 开始可以直接取模，不要多想

试题分析

讲一下各个题目易错点：

Day1 T1 注意下标从 0 开始可以直接取模，不要多想

Day1 T2 一定要记得离散化

讲一下各个题目易错点：

Day1 T1 注意下标从 0 开始可以直接取模，不要多想

Day1 T2 一定要记得离散化

Day1 T3 并查集要路径压缩，倍增优化注意细节

试题分析

讲一下各个题目易错点：

Day1 T1 注意下标从 0 开始可以直接取模，不要多想

Day1 T2 一定要记得离散化

Day1 T3 并查集要路径压缩，倍增优化注意细节

Day2 T2 权值线段树注意值域从 0 开始，怕被卡常也可以离散化一下

讲一下各个题目易错点：

Day1 T1 注意下标从 0 开始可以直接取模，不要多想

Day1 T2 一定要记得离散化

Day1 T3 并查集要路径压缩，倍增优化注意细节

Day2 T2 权值线段树注意值域从 0 开始，怕被卡常也可以离散化一下

Day2 T3 最短路千万不要写 SPFA，不要问我为什么

考试策略

考试策略 Day1

30min 读题

考试策略 Day1

30min 读题

T1 直接开，30min 快速完成

考试策略 Day1

30min 读题

T1 直接开，30min 快速完成

T2,T3 都需要一定的思维时间，可以先各花费 20min 把暴力写完

考试策略 Day1

30min 读题

T1 直接开，30min 快速完成

T2,T3 都需要一定的思维时间，可以先各花费 20min 把暴力写完

T2,T3 正解慢慢肝

考试策略 Day2

30min 读题

考试策略 Day2

30min 读题

T1 如果无法在 30min 中想到贪心先跳过

30min 读题

T1 如果无法在 30min 中想到贪心先跳过

T2 的 DP 非常明显，如果不敢写贪心的话 50min 快速调完线段树优化 DP 问题不大

30min 读题

T1 如果无法在 30min 中想到贪心先跳过

T2 的 DP 非常明显，如果不敢写贪心的话 50min 快速调完线段树优化 DP 问题不大

T3 先写个爆搜再看情况想 T1 或 T3

Thanks
