23 春季基础算法 B Contest09

ruogu

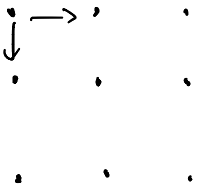
2023 年 4 月 22 日

题目概览

最小消耗

题目大意

给定一个 $n \times n$ 的格子图，每个格子有一个高度。初始你在 $(1,1)$，你要前往 (n,n)，你每次可以往上、下、左、右四个方向移动。一条路径耗费的体力值是路径上相邻格子之间高度差绝对值的最大值，问从起点走到终点的最小体力消耗值。
$(1 \le n \le 500, 1 \le h_{ij} \le 10^6)$



n^2 个点，

m 条边 $m \le 4n^2$，$\le 5\times10^5$

最小消耗

题目大意

给定一个 $n \times n$ 的格子图，每个格子有一个高度。初始你在 $(1,1)$，你要前往 (n,n)，你每次可以往上、下、左、右四个方向移动。一条路径耗费的体力值是路径上相邻格子之间高度差绝对值的最大值，问从起点走到终点的最小体力消耗值。

$(1 \le n \le 500, 1 \le h_{ij} \le 10^6)$

- 假设我们已知最小体力消耗值为 x，则对于任意两个相邻的点 (u,v)，我们都能判断能否由 u 走到 v：只需要满足 $|h_u - h_v| \le x$ 即可。

最大值是 x，我从 $(1,1)$ 到 (n,n) 所经过的边权都小于等于 x

二分

check(x) check(x+1)

check单调性，

最小消耗

题目大意

给定一个 $n \times n$ 的格子图，每个格子有一个高度。初始你在 $(1,1)$，你要前往 (n,n)，你每次可以往上、下、左、右四个方向移动。一条路径耗费的体力值是路径上相邻格子之间高度差绝对值的最大值，问从起点走到终点的最小体力消耗值。
$(1 \le n \le 500, 1 \le h_{ij} \le 10^6)$

- 假设我们已知最小体力消耗值为 x，则对于任意两个相邻的点 (u,v)，我们都能判断能否由 u 走到 v：只需要满足 $|h_u - h_v| \le x$ 即可。
- 因此可以二分答案，假设当前需要检查最小体力消耗值为 x 是否可行，我们从起点开始 DFS(或者 BFS)，过程中保证始终只经过满足 $|h_u - h_v| \le x$ 的 (u,v) 边即可。

咋check

并查集
BFS
dij ?

最小消耗

题目大意

给定一个 $n \times n$ 的格子图，每个格子有一个高度。初始你在 $(1,1)$，你要前往 (n,n)，你每次可以往上、下、左、右四个方向移动。一条路径耗费的体力值是路径上相邻格子之间高度差绝对值的最大值，问从起点走到终点的最小体力消耗值。
$(1 \le n \le 500, 1 \le h_{ij} \le 10^6)$

- 假设我们已知最小体力消耗值为 x，则对于任意两个相邻的点 (u,v)，我们都能判断能否由 u 走到 v：只需要满足 $|h_u - h_v| \le x$ 即可。
- 因此可以二分答案，假设当前需要检查最小体力消耗值为 x 是否可行，我们从起点开始 DFS(或者 BFS)，过程中保证始终只经过满足 $|h_u - h_v| \le x$ 的 (u,v) 边即可。
- 时间复杂度 $O(n^2 \times logH)$

最小消耗

- 甚至还可以继续优化算法。

```
check (x) {
    for m in M:
        if val[m] ≤ x :
            unite( ex[m], ey[m] )

    return findset(1,1) == findset(n,n)
```

最小消耗

- 甚至还可以继续优化算法。
- 首先将边权按从小到大排序，按排序顺序依次枚举每条边。

最小消耗

- 甚至还可以继续优化算法。
- 首先将边权按从小到大排序，按排序顺序依次枚举每条边。
- 在枚举到某条边时，使用并查集维护点的连通性，合并与该边相邻的两个点，并检查起点与终点此时是否处于同一集合中，如果已经处于同一集合中则当前边权即为答案，停止枚举；否则就继续枚举。

最小消耗

- 甚至还可以继续优化算法。
- 首先将边权按从小到大排序，按排序顺序依次枚举每条边。
- 在枚举到某条边时，使用并查集维护点的连通性，合并与该边相邻的两个点，并检查起点与终点此时是否处于同一集合中，如果已经处于同一集合中则当前边权即为答案，停止枚举；否则就继续枚举。
- 时间复杂度 $O(n^2)$

交换座位

题目大意

有 n 个座位，编号为 1 n。每次换座位时座位 i 上的人会换到作为 a_i 上，问至少要换多少次座位才能回到一开始的情形。

$(1 \le n \le 2 \times 10^5)$

$i \to a_i$

这题是找环

$i \to a_i \to a_{a_i} \to a_{a_{a_i}} \to \cdots \to$

7 $7k_0$

4 $4k_1$

k_2 1

$\mathrm{lcm}(7,4,1) = 28$

不是一个排列

交换座位

题目大意

有 n 个座位，编号为 1 n。每次换座位时座位 i 上的人会换到作为 a_i 上，问至少要换多少次座位才能回到一开始的情形。
$(1 \leq n \leq 2 \times 10^5)$

- 首先先只考虑如何计算答案。

求环长：

```
for (int i=0; i<n; i++) if (!vis[i]) {
    int x = a[i], c=1;  vis[i]=true;
    while ( x != i ) {
        ++c;  vis[x]=true;
        x= a[x]
    }
    cyc.push_back(c);
}
```

交换座位

题目大意

有 n 个座位，编号为 1 n。每次换座位时座位 i 上的人会换到作为 a_i 上，问至少要换多少次座位才能回到一开始的情形。
$(1 \leq n \leq 2 \times 10^5)$

- 首先先只考虑如何计算答案。
- 对于一个排列，其可以分割为若干个循环。如 $4\ 3\ 2\ 1\ 10\ 9\ 8\ 7\ 6\ 5$ 可以被分解为 $(1\ 4)(2\ 3)(5\ 10)(6\ 9)(7\ 8)$，显然一个循环内部经过该循环大小次操作将复位。

交换座位

题目大意

有 n 个座位，编号为 1 n。每次换座位时座位 i 上的人会换到作为 a_i 上，问至少要换多少次座位才能回到一开始的情形。
$(1 \leq n \leq 2 \times 10^5)$

- 首先先只考虑如何计算答案。
- 对于一个排列，其可以分割为若干个循环。如 $4\ 3\ 2\ 1\ 10\ 9\ 8\ 7\ 6\ 5$ 可以被分解为 $(1\ 4)(2\ 3)(5\ 10)(6\ 9)(7\ 8)$，显然一个循环内部经过该循环大小次操作将复位。
- 那么最终答案即为这若干个循环大小的最小公倍数。

交换座位

题目大意

有 n 个座位，编号为 1 n。每次换座位时座位 i 上的人会换到作为 a_i 上，问至少要换多少次座位才能回到一开始的情形。
$(1 \leq n \leq 2 \times 10^5)$

- 首先先只考虑如何计算答案。
- 对于一个排列，其可以分割为若干个循环。如 $4\ 3\ 2\ 1\ 10\ 9\ 8\ 7\ 6\ 5$ 可以被分解为 $(1\ 4)(2\ 3)(5\ 10)(6\ 9)(7\ 8)$，显然一个循环内部经过该循环大小次操作将复位。
- 那么最终答案即为这若干个循环大小的最小公倍数。
- 因此预处理出每个环的大小，然后依次遍历求 LCM 即可。

交换座位

题目大意

有 n 个座位，编号为 1 n。每次换座位时座位 i 上的人会换到作为 a_i 上，问至少要换多少次座位才能回到一开始的情形。
$(1 \leq n \leq 2 \times 10^5)$

- 首先先只考虑如何计算答案。
- 对于一个排列，其可以分割为若干个循环。如 $4\ 3\ 2\ 1\ 10\ 9\ 8\ 7\ 6\ 5$ 可以被分解为 $(1\ 4)(2\ 3)(5\ 10)(6\ 9)(7\ 8)$，显然一个循环内部经过该循环大小次操作将复位。
- 那么最终答案即为这若干个循环大小的最小公倍数。
- 因此预处理出每个环的大小，然后依次遍历求 LCM 即可。
- 然而有些情况答案将超过 long long 的范围，如何拿到满分呢？

交换座位

- 结论肯定不会变，那只能从优化求 LCM 这步入手。

4　　6　　9

$= 2^2$　　$2^1 \times 3^1$　　3^2

质因数　　2　　3

最大幂次　　2　　2

2^2　　$2^2 \times 3$　　$2^2 \times 3^2$

交换座位

- 结论肯定不会变，那只能从优化求 LCM 这步入手。
- 考虑如何求两个数的 LCM，可以对两个数质因子分解，对于所有出现过的质因子，最终对 LCM 的贡献为该因子在两个数质因子分解后的最高次幂。

交换座位

- 结论肯定不会变，那只能从优化求 LCM 这步入手。
- 考虑如何求两个数的 LCM，可以对两个数质因子分解，对于所有出现过的质因子，最终对 LCM 的贡献为该因子在两个数质因子分解后的最高次幂。
- 该过程也可以推广到求若干个数的 LCM。

交换座位

- 结论肯定不会变，那只能从优化求 LCM 这步入手。
- 考虑如何求两个数的 LCM，可以对两个数质因子分解，对于所有出现过的质因子，最终对 LCM 的贡献为该因子在两个数质因子分解后的最高次幂。
- 该过程也可以推广到求若干个数的 LCM。
- 因此可以考虑预处理所有数据范围内的素数，并对每个循环的大小进行质因子分解，那么最终答案即为所有质因子最高次幂的乘积，这样就可以顺利计算出模意义下的答案了。

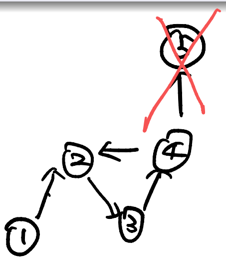
交换座位

- 结论肯定不会变，那只能从优化求 LCM 这步入手。
- 考虑如何求两个数的 LCM，可以对两个数质因子分解，对于所有出现过的质因子，最终对 LCM 的贡献为该因子在两个数质因子分解后的最高次幂。
- 该过程也可以推广到求若干个数的 LCM。
- 因此可以考虑预处理所有数据范围内的素数，并对每个循环的大小进行质因子分解，那么最终答案即为所有质因子最高次幂的乘积，这样就可以顺利计算出模意义下的答案了。
- 对于这种取模问题，需要注意运算时可能溢出 int 的这种小细节。

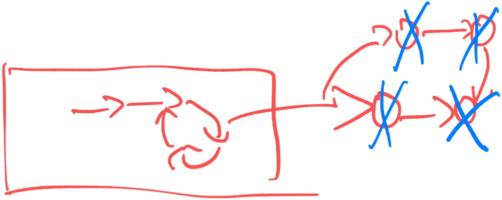
无尽的行走

题目大意

给定一个 n 个点 m 条单向边的图，问有多少个点满足从该点出发可以无限地在图中行走。
$(1 \le n \le 2 \times 10^5, 1 \le m \le 2 \times 10^5)$



⑤ 没有出度.



无尽的行走

题目大意

给定一个 n 个点 m 条单向边的图，问有多少个点满足从该点出发可以无限地在图中行走。
$(1 \le n \le 2 \times 10^5, 1 \le m \le 2 \times 10^5)$

- 考虑什么时候我们无法继续行走。

无尽的行走

题目大意

给定一个 n 个点 m 条单向边的图，问有多少个点满足从该点出发可以无限地在图中行走。
$(1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 2 \times 10^5)$

- 考虑什么时候我们无法继续行走。
- 当我们到达出度为 0 的某个点时，我们将无法继续行走。

无尽的行走

题目大意

给定一个 n 个点 m 条单向边的图，问有多少个点满足从该点出发可以无限地在图中行走。
$(1 \le n \le 2 \times 10^5, 1 \le m \le 2 \times 10^5)$

- 考虑什么时候我们无法继续行走。
- 当我们到达出度为 0 的某个点时，我们将无法继续行走。
- 因此我们可以不断将图中度数为 0 的点 v 删除，同时将指向 v 的所有点 u 的出度减一。重复该流程直至没有出度为 0 的点，此时剩下的所有点都是符合条件的。

无尽的行走

题目大意

给定一个 n 个点 m 条单向边的图，问有多少个点满足从该点出发可以无限地在图中行走。
$(1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 2 \times 10^5)$

- 考虑什么时候我们无法继续行走。
- 当我们到达出度为 0 的某个点时，我们将无法继续行走。
- 因此我们可以不断将图中度数为 0 的点 v 删除，同时将指向 v 的所有点 u 的出度减一。重复该流程直至没有出度为 0 的点，此时剩下的所有点都是符合条件的。
- 上述流程实际上就是一个逆向的拓扑排序，实现时可以考虑建图时将原图的所有边反向，不断删除入度为 0 的点 (此时这个点在原图对应出度为 0)，最终无法操作时即可得到答案。

无尽的行走

题目大意

给定一个 n 个点 m 条单向边的图，问有多少个点满足从该点出发可以无限地在图中行走。
$(1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 2 \times 10^5)$

- 考虑什么时候我们无法继续行走。
- 当我们到达出度为 0 的某个点时，我们将无法继续行走。
- 因此我们可以不断将图中度数为 0 的点 v 删除，同时将指向 v 的所有点 u 的出度减一。重复该流程直至没有出度为 0 的点，此时剩下的所有点都是符合条件的。
- 上述流程实际上就是一个逆向的拓扑排序，实现时可以考虑建图时将原图的所有边反向，不断删除入度为 0 的点 (此时这个点在原图对应出度为 0)，最终无法操作时即可得到答案。
- 时间复杂度 O(M + N)。

三色图

题目大意

给定一个 n 个点 m 条边的无向图，每条边有一个边权以及颜色 (红、绿、蓝)。问能否选择 k 条边满足要么颜色只包含绿色和红色，要么颜色只包含绿色和蓝色，且 n 个点之间两两联通，如果能的话返回可行方案中的最小边权和。
$(1 \le n \le 10^5, n-1 \le m \le 3 \times 10^5)$

三色图

题目大意

给定一个 n 个点 m 条边的无向图，每条边有一个边权以及颜色 (红、绿、蓝)。问能否选择 k 条边满足要么颜色只包含绿色和红色，要么颜色只包含绿色和蓝色，且 n 个点之间两两联通, 如果能的话返回可行方案中的最小边权和。
$(1 \le n \le 10^5, n-1 \le m \le 3 \times 10^5)$

- 题目给出了两种限制，我们可以对每种限制分别求一个答案，取 Min 即为最终答案。

三色图

题目大意

给定一个 n 个点 m 条边的无向图，每条边有一个边权以及颜色 (红、绿、蓝)。问能否选择 k 条边满足要么颜色只包含绿色和红色，要么颜色只包含绿色和蓝色，且 n 个点之间两两联通, 如果能的话返回可行方案中的最小边权和。
$(1 \leq n \leq 10^5, n-1 \leq m \leq 3 \times 10^5)$

- 题目给出了两种限制，我们可以对每种限制分别求一个答案，取 Min 即为最终答案。
- 然而枚举 k 显然是不现实的，我们能否一次性将所有 k 的答案都求出来呢？

三色图

- 首先假如 $k = n - 1$，那答案可以通过做一次 kruskal 统计。、

三色图

- 首先假如 $k = n - 1$，那答案可以通过做一次 kruskal 统计。、
- 将边按边权从小到大排序，并用并查集维护点的连通性。遍历每一条边，如果不符合颜色要求则跳过，如果与该边相关联的两个点不在同一连通块内，则合并这两个点所在的连通块, 标记这条边为已使用，并统计这条边的贡献。

三色图

- 首先假如 $k = n - 1$，那答案可以通过做一次 kruskal 统计。、
- 将边按边权从小到大排序，并用并查集维护点的连通性。遍历每一条边，如果不符合颜色要求则跳过，如果与该边相关联的两个点不在同一连通块内，则合并这两个点所在的连通块, 标记这条边为已使用，并统计这条边的贡献。
- 那么对于其余的 k, 我们能否利用已有的生成树帮助我们求出答案？

三色图

- 首先假如 $k = n - 1$，那答案可以通过做一次 kruskal 统计。、
- 将边按边权从小到大排序，并用并查集维护点的连通性。遍历每一条边，如果不符合颜色要求则跳过，如果与该边相关联的两个点不在同一连通块内，则合并这两个点所在的连通块，标记这条边为已使用，并统计这条边的贡献。
- 那么对于其余的 k，我们能否利用已有的生成树帮助我们求出答案？
- 我们可以在已有的生成树上按边权从小到大依次加上符合颜色要求的未使用的边。

三色图

- 首先假如 $k = n - 1$，那答案可以通过做一次 kruskal 统计。、
- 将边按边权从小到大排序，并用并查集维护点的连通性。遍历每一条边，如果不符合颜色要求则跳过，如果与该边相关联的两个点不在同一连通块内，则合并这两个点所在的连通块, 标记这条边为已使用，并统计这条边的贡献。
- 那么对于其余的 k, 我们能否利用已有的生成树帮助我们求出答案？
- 我们可以在已有的生成树上按边权从小到大依次加上符合颜色要求的未使用的边。
- 需要注意的是如果 $k = n - 1$ 时无解，那么后续更大的 k 也同样无解。

三色图

- 首先假如 $k = n - 1$，那答案可以通过做一次 kruskal 统计。、
- 将边按边权从小到大排序，并用并查集维护点的连通性。遍历每一条边，如果不符合颜色要求则跳过，如果与该边相关联的两个点不在同一连通块内，则合并这两个点所在的连通块，标记这条边为已使用，并统计这条边的贡献。
- 那么对于其余的 k，我们能否利用已有的生成树帮助我们求出答案？
- 我们可以在已有的生成树上按边权从小到大依次加上符合颜色要求的未使用的边。
- 需要注意的是如果 $k = n - 1$ 时无解，那么后续更大的 k 也同样无解。
- 时间复杂度 O(N + MlogM)

The End

谢谢大家