

NOIP 模拟赛 1 题解

Contents

1	T1	3
1.1	问题 1	3
1.1.1	描述	3
1.1.2	解法	3
1.2	问题 2	3
1.2.1	描述	3
1.2.2	解法	3
1.3	原问题	3
1.3.1	描述	3
1.3.2	解法	3
2	T2	5
2.1	结论	5
2.2	证明	5
2.2.1	结论 1	5
2.2.2	结论 2	5
2.2.3	结论 3	5
2.3	解法	6
3	T3	7
4	T4	8
4.1	做法 1	8

4.2	做法 2	8
4.3	做法 3	8
4.4	做法 4	8

1 T1

首先这个最短路问题可以跑 Dijkstra，但其实我并不想让 Dijkstra 过的。

注意到边权只有 2 种，所以我们有更优的做法。

我们回顾一下我们会的最短路做法：

1.1 问题 1

1.1.1 描述

边权固定。

1.1.2 解法

用 queue 维护。

每次把松弛的点 push 进去。

1.2 问题 2

1.2.1 描述

边权只有 01。

1.2.2 解法

用 deque 维护：

- 如果是 0 则 push_front;
- 如果是 1 则 push_back。

1.3 原问题

1.3.1 描述

抽象成图论问题，即是边权种类数较少。

1.3.2 解法

注意到 1.1, 1.2 其实本质上在模拟 Dijkstra 的优先队列。

那么本题两种边权，我们一样可以这样做。

我们维护两个队列，分别表示两种不同边权的松弛。每次我们比较两个队列队头取较小的那个，就可以了。

这样我们能保证每次取最小的，同时，不难发现两个队列内最短路长度分别单调，所以是正确的！

时间复杂度 $O(nmk^2)$ ，数据乱造的不知道能不能卡掉 Dijkstra，卡不掉就当 T1 送分吧。

2 T2

2.1 结论

首先给出这道题的贪心策略：如果能移动身体则移动身体，否则把所有脚按照从右到左的顺序依次移到尽量右的位置。

2.2 证明

2.2.1 结论 1

2.2.1.1 描述

如果能移动身体则一定移动身体。

2.2.1.2 证明

因为这是一步迟早要做的必要操作，并且只会有利于所有脚的右移。

2.2.2 结论 2

2.2.2.1 描述

如果要移动一只脚，那么一定将它移动到尽量靠右的位置。

2.2.2.2 证明

因为这样有利于身体和在它后面的脚右移，而对于在它前面的脚不造成影响。

2.2.3 结论 3

2.2.3.1 描述

如果身体不能移动，一定要从右到左移动所有脚后才能让身体再次可以移动。

2.2.3.2 证明

由结论 2，任意时刻所有脚都是站在一段连续的非破损位置。所以一开始只有最右边的脚可以移动，接下来是第二右的……以此类推。

2.3 解法

那么我们只要实时维护身体的位置、目前最左边的脚的位置、目前身体下方前 a 个非破损位置。其中目前身体下方前 b 个非破损位置可以用队列维护。

对于每个时刻，如果可以移动身体（即移动后最左边的脚还在身体下放）则移动身体，同时更新队列；否则如果可以移动脚则移动脚；都不能就直接输出无解。

时间复杂度 $O(n)$ 。

3 T3

利用一点概率论知识，我们可以知道对于每个点 u ，统计 c_u 代表有多少以它为端点的路径价值 $< t$ ，那么答案概率就是 $1 - \prod_{u=1}^n \frac{c_u}{n}$ 。们来算 c_u 就行了。

不妨以 1 为根。记 \bar{w}_u 表示从 u 到根的路径上的权重异或。很容易发现 $(u \rightarrow v)$ 的路径的权值就是 $\bar{w}_u \oplus \bar{w}_v \oplus w_{\text{LCA}(u,v)}$ ，其中 LCA 表示最近公共祖先。

于是我们自然在 LCA 处统计路径。考虑如果一个点 r 有两个子树，我们要考虑跨过这两个子树的路径，那么直接把其中点数较多的一个建 01 Trie，点数较小的一个，枚举其中每个点 u 放到 01Trie 上查询与 $\bar{w}_u \oplus w_r$ 异或值 $< t$ 的点的个数，同时给这些 Trie 中的点打上 tag 来加 1。

这样有什么好处？每个点可以继承他大小最大的儿子的 Trie 树，把其他点暴力放到 Trie 树上查询，并在查询完之后插入 Trie 中，这样可以省下来建立重儿子的 Trie 的时间。

我们注意到每个点如果被暴力插入，那么一定代表着此时所在的子树大小再往父亲跳一步会翻倍，所以每个点最多需要暴力插入/查询 $\log n$ 次。每次操作复杂度 $\log w$ 。从而总复杂度为 $O(n \log n \log(\max w_i))$ 。

其中 Trie 树上的操作和计数可能有些细节，需要当心不要重复统计了。

4 T4

4.1 做法 1

枚举答案矩形的右边界，枚举答案的左边界，按 y 方向从小到大枚举，只需要记录上一次 y 最大值，因为上下边界必定相邻。

时间复杂度： $O(n^3)$ 。

期望得分：40。

4.2 做法 2

枚举一个边界，考虑其他点的影响， y 小于的点影响下边界， y 大于的点影响上边界，一直枚举，要么它不在上下边界上，那么一定在右边界上，更新答案。

期望得分：60。

4.3 做法 3

不再考虑枚举边界，直接考虑最后答案会是什么情况。考虑分治，枚举右边界，使用数据结构维护上下边界，通过上下边界在数据结构上求最小值，更新答案。

数据结构：

- 插入，查前驱后继：可以使用 `set`。考虑到不删除性，可以用并查集。
- 查区间最小值。

时间复杂度 $O(n \log^2 n)$ 。

期望得分：80。

4.4 做法 4

描述：答案必过中线。使用公式 $2 \times \max(w, h) + 2$ 来直接计算可能的最大值。

时间复杂度 $O(n \log n)$ 。

期望得分：100。