

PION 2024 模拟赛题解

Moeebius

题目名称	冒泡	山路	机器	旅行
目录	pop	road	machine	tour
原题	QOJ 3044	gym 100917F	QOJ 3040	某场梦熊的 T2
题解	Here	Here	Here	Here

提交源程序文件名

对于 C++ 语言	pop.cpp	road.cpp	machine.cpp	tour.cpp
-----------	-------------------------	--------------------------	-----------------------------	--------------------------

编译选项

对于 C++ 语言	<code>-O2 -std=c++14 -static</code>
-----------	-------------------------------------

冒泡 (pop)

【算法 1】

各种暴力，视复杂度可以获得 5/15/35 分。

【算法 2】

考虑到最大的 value 是好算的： $\sum_{i=1}^N |i - (N - i + 1)|$ ，当 N 为偶数时，这等价于要让 A 中不超过 $\frac{N}{2}$ 的数与 B 中大于 $\frac{N}{2}$ 的数两两匹配。

有经典结论：交换相邻两项，让一个序列变为有序的最少操作次数等于该序列的逆序对数。因此不难发现，记 $S = \{i \mid B_i \leq \frac{N}{2}\}$, $T = \{i \mid B_i > \frac{N}{2}\}$ ，则重排之后得到的序列 A' 一定是将 A 中 $\leq \frac{N}{2}$ 的元素依次填入 T 中位置，将 A 中 $> \frac{N}{2}$ 的元素依次填入 S 中位置，所得到的序列。

操作次数相当于求 01 序列逆序对，可以直接计算。时间复杂度 $O(n)$ ，可以通过子任务 4，加上暴力期望得分 50。

【算法 3】

考虑 N 是奇数的情况，此时 A 中 $\frac{N+1}{2}$ 可以和 B 中任意数匹配。这也是好做的：

1. A 中 $\leq \frac{N-1}{2}$ 与 B 中 $> \frac{N+1}{2}$ 匹配；
2. A 中 $\leq \frac{N+1}{2}$ 与 B 中 $\geq \frac{N+1}{2}$ 匹配。

只有这两种可能，都跑一遍算法 2 取操作次数最小值即可。时间复杂度 $O(n)$ ，可以通过本题。

山路 (road)

【算法 1】

我会基环树找环!

可以通过子任务 1, 期望得分 10。

【算法 2】

我会状压!

可以通过子任务 2, 期望得分 10。

【算法 3】

我会最短路!

考虑枚举每个点 p 。

注意到所有包含这个点的最小边简单环都形如 $p \rightarrow u \rightsquigarrow v \rightarrow q$, 其中 $u \rightsquigarrow v$ 是删除 p 及其所有出边后, u 到 v 的最短路。

枚举 p 后暴力跑全源最短路, 复杂度 n^4 左右, 可以通过子任务 3 (实现优秀可以通过子任务 4), 期望得分 20/40。

【算法 4】

我会分治!

考虑记 $f(l, r, i, j)$ 表示保留 $[1, l] \cup (r, n]$ 内所有点时, i 到 j 的最短路。

分治处理, 到叶子时计算答案。否则取 $\text{mid} = \frac{l+r}{2}$, 先用类似 Floyd 的方式加入 $(\text{mid}, r]$ 点的贡献并递归 (l, mid) , 然后回滚, 再加入 $[l, \text{mid}]$ 的贡献并递归 $(\text{mid} + 1, r)$, 然后回滚。

时间复杂度 $O(n^3 \log n)$, 空间复杂度 $O(n^2 \log n)$, 可以通过本题。

【算法 5】

我会找性质!

考虑枚举点 p 并建出以 p 为根的最短路树, 考虑枚举每条非树边 (u, v) , 如果 (u, v) 的 LCA 不是 p 则一定不优。否则用 $\text{dis}_u + \text{dis}_v + w_{u,v}$ 更新答案。

时间复杂度 $O(n^3 + n^2 \log n)$, 可以通过本题。

机器 (machine)

【算法 1】

我会状压！可以通过子任务 1，期望得分 10。

【算法 2】

考虑发掘一些性质。显然，只要让所有蓝色弹珠归位，就一定完成了重排。

不难发现，只有以下操作是有用的：

1. $[BR] \xleftrightarrow{C+3} [RB]$ 。(效果：蓝色弹珠移动一格)
2. $[BRR] \xleftrightarrow{C+4} [RRB]$ 。(效果：蓝色弹珠移动两格)
3. $[BBR] \xleftrightarrow{C+5} [RBB]$ 。(效果：蓝色弹珠移动两格，并越过一个蓝色弹珠)

注意到我们不会选择越过一个奇偶性与自己相同的蓝色弹珠，因为这样一定不优。

此外，我们还能发现，两次操作 1 一定劣于一次操作 2。因此，我们如果要将一个蓝色弹珠移动 d 格，所需代价就是 $\lfloor \frac{d}{2} \rfloor (C+4) + (d \bmod 2)(C+3) + \Delta$ ，其中 Δ 表示需要越过的蓝色弹珠数量。

状压 DP，记 $f_{i,S}$ 表示考虑到 A 序列从左往右第 i 个蓝色弹珠， B 序列 S 集合中的蓝色弹珠已被匹配的最小代价。

具体来说，和 d 有关的贡献可以在转移时计算；记 p_i 为 A 中从左到右第 i 个蓝色弹珠的终点位置， $\sum \Delta = \sum_{i < j} [p_i > p_j]$ 。

可以通过子任务 2，期望得分 30。

【算法 3】

一种网络流做法，由于与正解关系不大所以略去，感兴趣的同学可以参考 [原题题解](#)。期望得分 50。

【算法 4】

考虑特殊性质 B：所有蓝色弹珠位置奇偶性都相同。此时不难发现按顺序匹配一定不劣。可以通过子任务 4，期望得分 15。

【算法 5】

考虑拓展算法 4。注意到将 A 中蓝色弹珠按照终点位置的奇偶性分成 S, T 两组，组内一定是按照顺序匹配的。证明并不困难：

- 由于终点奇偶性都相同， $\sum (d \bmod 2)(C+3)$ 是固定的；
- 按顺序匹配，剩下部分的代价一定不劣。□

因此我们可以记 $f_{i,j}$ 表示「 A 中前 i 个蓝色弹珠有 j 个终点为奇数位置」的最小代价，经过必要的预处理后可以在 $O(n^2)$ 时间内完成转移，可以通过本题。

【一些细节】

注意最后构造方案的时候不能按照任意顺序挪动蓝色弹珠。考虑如下例子：

BRRBRRR → RRRBRRB

如果先挪左边的弹珠，再挪右边的弹珠，则会带来额外 1 的代价。

对于特殊性质 C，我们可以从右往左移动弹珠；对于一般的情况，我们将所有弹珠分成「向右移动」、「向左移动」、「不动」三类，第一类从右往左挪，第二类从左往右挪，第三类不管。易证正确性。

当然，如果你没有观察到这一点，也可以通过拓扑排序构造方案。

旅行 (tour)

【算法 1】

我会暴力!

可以通过子任务 1, 期望得分 10。

实现优秀的话还能通过子任务 7, 期望得分 20。

【算法 2】

我会树剖!

不存在 4 操作的测试点是树剖板子。可以通过子任务 4, 期望得分 15。

【算法 3】

我会拆邻域!

如果 $k = 1$, 那么操作具有交换律和结合律, 考虑使用树剖完成操作 2,3, 只需要考虑邻域加法。

应用 洒水器 一题的 Trick: 「 x 的 k 级祖先 p 的 r 级儿子」和「 x 的 $k+1$ 级祖先 q 的 $r-1$ 级儿子」只相差 p 的 r 级和 $r-1$ 级儿子, 因此从 x 往上跳的时候每个位置只需要修改 $O(1)$ 层。

对于每个点 x 维护 $\text{tag}_{x,i}$ 表示 x 的第 i 级儿子的加法标记。查询和修改的时候往上跳不超过 10 层即可。可以通过子任务 6, 配合算法 2 期望得分 30。

【算法 4】

考虑延续算法 3 的思想。由于操作不能交换我们无法找出直接维护 tag 了, 并且由于有链和子树操作我们只能使用 DFS 序。

考虑把「 x 的 i 级儿子全部执行某个操作」转化为: x 子树内 $\text{dep} = j$ 的所有点执行操作。将 j 放到线段树的 tag 里。注意到 tag 表示整段区间都要执行某个操作, 又因为 $j \leq \text{dep}_x + r$, 我们发现每个线段树节点上最多只有 $O(r)$ 个 tag!

注意到我们在节点上还要维护不限制深度的 tag 来支持链和子树的操作, 实现略有细节。时间复杂度 $O(qr^2 \log n + qr \log^2 n)$, 由于卡不满且常数较小, 可以通过本题。