SSZX NOIP2021 模拟赛解题报告

（2021.10.16　8:00～12:30）

1. 画图

题目大意：如果两个点之间有不相交的两条路径，而且没有边相连，那么可以连一条边，问最多可以添加到多少条边。

1.1 算法 1

按照题意模拟，需要使用搜索等算法。

时间复杂度$O(?)$。

期望得分$5-15$分。

1.2 算法 2

观察到这两条路径其实是一个环，如果一个点添加了和环上与之距离为2的点之间的边，那么可以获得一个新的环，在新的环上，这个点与原来在环上距离为3的点的之间的距离就变成2了。因此我们可以使每次连接的两个点的其中一条路径长度为2，查看能否找出另一条路径。

维护去掉每一个点之后图的连通性，每次枚举可以添加哪一条边，可以做到$O(n^4)$或者更优的复杂度。

期望得分$25-40$分。

1.3 算法 3

发现一个点双连通分量之内点会连成完全图，因此只需要找出点双连通分量即可。暴力为$O(n^2)$，使用 tarjan 算法为$O(n+m)$。

期望得分100分。

2. 交换

题目大意：给定一个排列，每次给定一个区间，交换区间内的两个数使得排列字典序最小，询问交换的位置，多次询问。

2.1 算法 1

按照题意模拟，枚举交换位置并比较。

时间复杂度$O(mn^3)$。

期望得分20分。

2.2 算法 2

不难发现给定区间之外的位置对每个询问的答案无影响，所以每次的问题就是取出一个子段，问这个子段怎样交换一次字典序最小。

根据字典序定义，我们需要找到最小的位置满足通过交换可以使这个位置变小，也就是说这个位置不是后缀最小值，因此从后往前取最小值，找出可以变小的位置中最靠前的一个。最后与把这个位置与这个位置之后的最小值交换就是最优的了。

时间复杂度$O(mn)$。

期望得分$40-50$分。

2.3 算法 3

对于性质 A 可以用 set 暴力找出这些逆序对。因为每次交换的时候一定会使逆序对减少，所以对于每个询问，枚举哪些逆序对在区间中，选择最优的交换，并更新减少的逆序对。

时间复杂度$O(nlog(n)+100m)$

结合前面的算法，期望得分$55-60$分。

2.4 算法 4

对于性质 B 可以发现每个区间第一个可以交换变优的位置会很靠前。直接暴力枚举前几位看一下是不是能交换，用线段树维护区间最小值。应该可以取得不错的效果。

结合前面的算法，期望得分$65-70$分。

2.5 算法 5

问题在于如何求出第一个能变小的位置。可以找出这一段区间从开头开始的最长的连续上升段，那么交换的一定是连续上升段内和连续上升段后的数字。可以求出连续上升段之后的最小值，然后找到连续上升段中第一个比这个最小值大的位置，交换这两个位置就是最优的。

求连续上升段可以为每个位置维护一个标记，表示这个位置是否比下一个位置大。使用线段树二分查找第一个有标记的位置就能找到最长连续上升段。线段树维护最小值很简单。最后查询连续上升段中比一个数大的最小位置，这个可以维护区间的最大值，同样二分查找即可。

时间复杂度O(nlog(n))

期望得分100分。

3. 步行

题目大意：一棵树，求一个每个点出现特定次数的序列，使得这个序列相邻的两位之间，包括最后一位与第一位之间的距离之和最大。每次去掉一条边加上一条边询问。

3.1 算法 1

暴力枚举游览序列，查询树上距离。

时间复杂度$O(S!nm)$。

期望得分8分。

3.2 算法　2

考虑树形 dp，令$f_{i,j}$表示i号点的子树里面的所有点在序列中构成j个连续段，子树内步行距离和最大值。转移时枚举两个子树，以及有多少段合并起来了。

时间复杂度$O(mS^3)$。

期望得分16分。

3.3 算法 3

观察到每条边走的次数是把这条边断掉之后，出现次数和较小的连通块的出现次数和的两倍。

答案显然不可能超过这个值，构造也比较容易，找到树的带权重心，权为每个点出现次数，这样所有边断掉之后较大的连通块都包含重心。接下来就是把重心去掉，每个部分的出现次数之和都不超过$\frac{S}{2}$，容易构造出一种序列使得相邻两位都不来自相同的连通块，这样就达到了这个最大值。

既然有了这个结论，每次 dfs 算出子树和就可以得到答案了。

时间复杂度$O(nm)$

期望得分$36-44$分。

3.4 算法 4

可以发现，如果按照每条边来算贡献的话，那么在添加的道路两个端点对应原树的路径上的边贡献会改变，其他的均不会改变。因为树是完全二叉树，两点之间边数只有$O(\log(n))$级别，暴力计算即可。

时间复杂度$O(n+m\log(n))$

结合前面的算法，期望得分$44-52$分。

3.4 算法 5

如果树是一条链，那么修改一条边之后也只会是一个““T”状图。在这个“T”状图上二分重心的位置，然后计算对应部分的贡献。

时间复杂度$O(n+m\log(n))$。

结合前面的算法，期望得分$52-64$分。

3.6 算法6

首先当n较大的时候，信息错误使用倍增 lca 代替暴力判断即可 。

信息正确时，只需要延续算法 5 的思路，不妨把新添加的边端点在原树上的对应链提出来，加上新添加的边构成一个环。整棵树就变成了环套树。把环以外的所有边的贡献先算出来，这个可以通过子树和的方式计算。然后把环外面每个点的出现次数都加到对应的环上的点上面。这样问题就变成了一个环断掉一条边，答案是多少。采用算法 5 的二分方法，不过在这里是使用倍增数组进行二分。实现良好的话可以通过。

时间复杂度$O((n+m)\log(n))$。

期望得分$68-100$分。

3.7 算法 7

可以发现在题目中所有的询问都有类似的形式，就是对于一条从下往上的链，链上每条边对应的子树的大小都增加一个固定值v，然后根据子树大小来算边的贡献之和。因此可以对于v进行排序，由大到小处理，每个点都会逐渐出现（子树大小$+v$）$< \frac{S}{2}$的情况，而且这个情况是由下向上出现的。而我们的二分过程就是二分最高的$< \frac{S}{2}$的位置，因此可以使用并查集代替二分。

而且v只会是$S - size_i, -size_i, size_i$三种之一，其中$size_i$为原树中子树$i$的出现次数和。对$n$个size进行排序，使用$O(n\log(n))$，$O(1)$lca，以及按秩合并路径压缩并查集，可以得到复杂度为$O(n\log(n) + m\alpha(n))$的做法。

期望得分100分。

4. 航行

题目大意：长为 n 的数轴，一个点在其中运动，每个位置有一定的概率使得点向右加速，一定概率向左加速，问这个点期望何时离开数轴，对每一个初始位置都计算。

4.1 算法 1

输出 1。

时间复杂度$O(1)$。

期望得分4分。

4.2 算法 2

根据样例解释的方式解方程，$2-4$手算难度依次增大。

期望得分$8-16$分。

4.3 算法 3

对于所有$p_i \in \{0,100\}$的点，所有出发的点路径确定，建立点数n^2的有向图，dfs 即可。

时间复杂度$O(n^3)$或$O(n^2)$

结合前面期望得分32分。

4.4 算法 4

既然有了解方程的做法，而且是线性的，不难想到高斯消元，令$x_{i,j}$表示点i速度为j的期望步数，对于随机的情况可以认为-1不存在，直接消元n次复杂度为$O(n^7)$。

期望得分$28-36$分，结合前面得到$40-48$分。

4.5 算法 5

发现算法 4 中n次消元系数都是相同的，只有右边的常数不同，于是直接在矩阵右边添加n列一同消元。

时间复杂度$O(n^6)$。

期望得分40分，结合前面得到52分。

4.6 算法 6

很多状态是没用的，可以发现如果速度达到$\sqrt{2n}$左右的时候，这个点一定走出数轴了。因此状态数变为$O(n^{1.5})$，再进行消元。

时间复杂度$O(n^{4.5})$。

期望得分52分，结合前面得到60分。

4.7 算法 7

考虑处理-1的问题，不妨把不为0的转移建成有向图，如果一个强连通分量不存在任何出边或者有出边通向存在-1的连通分量，那么这个强连通分量的答案也会是-1，否则答案就不是-1，缩点之后按照拓扑序处理，然后删去所有-1的状态。

这个时候所有不是-1的状态都不会进入是-1的状态，再进行消元即可。

结合前面期望得分72分。

4.8 算法 8

在消元过程中，我们一行最多只会有三个位置有值，很浪费。考虑减少未知数的数量。一种优化方式是，考虑算出一个静止状态到另一个静止状态的转移概率，两个静止状态之间，由于速度连续，所以一定是一直往左或者一直往右的，这个简单的三维 dp 就可以算出，这一部分为$O(n^{2.5})$。再利用高斯消元算出每个静止状态出现的期望次数，通过 dp 的结果可以求出所有状态的期望出现次数之和，这个与答案是相同的。静止的状态只有n种，消元复杂度为$O(n^3)$。同样采用算法 7 的判断-1的方式。

时间复杂度$O(n^3)$

期望得分100分。