NOIP2019 模拟

Day1 题解

一. 题目概况

题目名称	achen	tree	easy
可执行文件名	achen	tree	easy
输入文件名	achen.in	tree.in	easy.in
输出文件名	achen.out	tree.out	easy.out
时间限制	1.0s	1.0s	1.0s
空间限制	64MB	256MB	256MB
测试点数量	20	打包测试	20
单个测试点分值	5	见题面	5
题目类型	传统	传统	传统

二. 可执行文件名需加后缀

对于 C++语言	achen.cpp	tree.cpp	easy.cpp
对于 C 语言	achen.c	tree.c	easy.c
对于 Pascal 语言	achen.pas	tree.pas	easy.pas

三. 编译选项

对于 C++语言	-lm	-lm	-lm
对于 C 语言	-lm	-lm	-lm
对于 Pascal 语言			

四. 注意事项

1. 文件名(程序名和输入输出文件名)必须使用英文小写。
2. 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
3. C/C++中函数 main()的返回值必须是 int，程序正常结束时返回值必须是 0。

1 achen

1.1 题意简述

求有多少个 1 到 n 的排列，满足 P[1]=A、P[n]=B，且 abs(P[i]-P[i+1])<=2。

1.2 算法 1

n!枚举，暴力判断。期望得分 30。

1.3 算法 2

可以发现，当 A=1,B=n 时，有递推式 f[i]=f[i-1]+f[i-3]，表示一次走一步或者一次走两步后退回一步再一次走两步。结合算法 1 期望得分 60。

1.4 算法 3

这种情况只可能有一种走法。输出 1 即可。结合算法 1 期望得分 40。

1.5 算法 4

考虑合并算法 2 和算法 3。我们设 A<B，当 A!=1 时我们必须先从 A 走到<A 的点然后走到 A+1，这段只有一种走法，同理，当 B!=n 时，最后一定是走到了 B-1，然后走过>B 的点最后走到了 B，对于中间那一段用算法 2 的方法 dp 就行了。需要预处理 dp 值。

复杂度 O(n+T)，期望得分 100。

2 tree

2.1 题意简述

一棵 n 个点的树，找一个根和一个深度最大的点，这个点的子树需包含所有颜色。

2.2 算法 1

暴力枚举根，f[i][j]表示 i 这个点的子树是否含有 j 这个颜色，暴力转移。

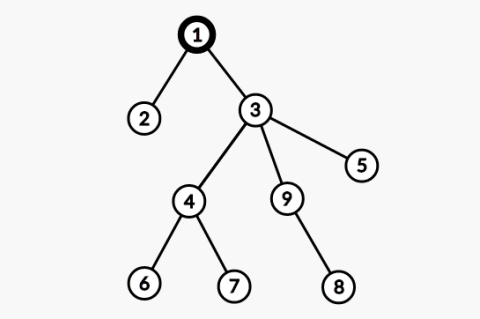
时间复杂度$O(n^2m)$，期望得分 20。

2.3 算法 2

用 dp 或者 dfs 直接找最长链。结合算法 1 期望得分 30。

2.4 算法 3

先随便找一个点作为临时的根，然后对于我们枚举到的一个点，我们考虑它作为 LCA 的时候的最优答案，显然真正的根可能会是他某个儿子的子树里的一个点，或者他自己，或者，不在他的子树内，对于这几种情况，我们都算一个最远的那个点离他多远，把最远的那个作为根就可以了。那么我们还要算，当根是在那里的时候，这个点的子树合不合法(有没有包含每种颜色的点)。



比如，右边这个图，当真正的根是 6，当前点是 3 的时候，我们就需要知道，他的子树的点集 T={1,2,3,5,9,8}这个集合里面是否包含了每种颜色的点。

我们用 f[i][j]计算 i 这个点的子树包含多少个 j 这种颜色的点。

那么我们就可以用 f[1]-f[4]算得 T 中包含的每种颜色的点的个数。

我们对于每个点，枚举根在哪个方向，算这个方向上最远的点，复杂度是 O(n)的。算 f[i][j]的总复杂度是 O(nm)的，枚举根方向再算合不合法，复杂度也是 O(nm)。

所以总复杂度是 O(nm)。期望得分 60。

2.5 算法 4

发现上一个做法的瓶颈在于算 f[i][j]。就是判断一个子树是不是包含所有颜色的点，和判断一个反树(整个树减去一个子树)不是包含所有颜色的点。

对于第一个问题，把同一种颜色的点按照 dfs 序排序，每个点到根的路径+1，相邻两个点 LCA 到根的路径-1，然后就算出了一个点的子树所包含的颜色种数，用树上差分优化一下。实现时并不需要排序，只要按照 dfs 从 1 到 n 扫一遍，随时维护每种颜色上一个出现位置即可。

对于第二个问题，我们可以反过来考虑，既然反树要包含所有颜色的点，那么那个子树就不能把某种颜色的点全部包含完，对于一个颜色，所有这种颜色的点的 LCA 及 LCA 的祖先，都是包含完这种颜色的，对于每种颜色的所有点的 LCA 位置打个标记，然后从下到上传一遍标记就可以了。

可以用倍增求 LCA，常数不太大是可以通过此题的，当然也可以用 tarjan 求 LCA。期望得分 100。

3 easy

3.1 题意简述

有一个数组 a，求有多少对(l,r)满足$1 \le l \le r \le n$，且把$a[l], a[l+1], \ldots, a[r]$排序后相邻数差的绝对值不超过 1。

3.2 算法 1

枚举 l 和 r，暴力排序判断。复杂度 $O(n^3 logn)$，期望得分 10。

3.3 算法 2

枚举 l，r 从 l 到 n 枚举，每次添加一个数进去，维护最大值、最小值、一共出现了多少个不同的数。复杂度 $O(n^2)$，期望得分 30。

3.4 算法 3

对于一段区间[l, r]，设最大值和最小值分别为 A、B，那么当 A-B=r-l 时这个区间满足条件，而对于所有区间都满足 A-B+l≥r，我从小到大枚举 r，在线段树里放对于每个 l 的 A-B+l，线段树里需要维护区间最小值，和取得最小值的位置个数。

关于如何动态维护 A-B+l，我们考虑每次 r 向右移动，每个区间应该加入 a[r]，A 可能会由原来的最大值变为 a[r]，B 可能会由原来的最小值变为 a[r]，我们考虑在移动 r 时，同时维护两个单调栈，一个单减，用来处理最大值 A，另一个单增，用来处理最小值 B，在弹栈的时候顺便在线段树上区间修改就可以解决这个问题。

时间复杂度 O(nlogn)，结合算法 2 期望得分 60。

3.5 算法 4

在上个子任务的基础上，我们考虑如果有相同的a_i时，式子变成了 A-B+cnt=r-l，cnt 表示某些位置上的数不是在区间内第一次出现的个数。考虑 r 向右移动时，对于每个 l，cnt 的改变量。预处理出 a[r]上一次出现的位置为 last[r]即可。

时间复杂度 O(nlogn)，期望得分 100。