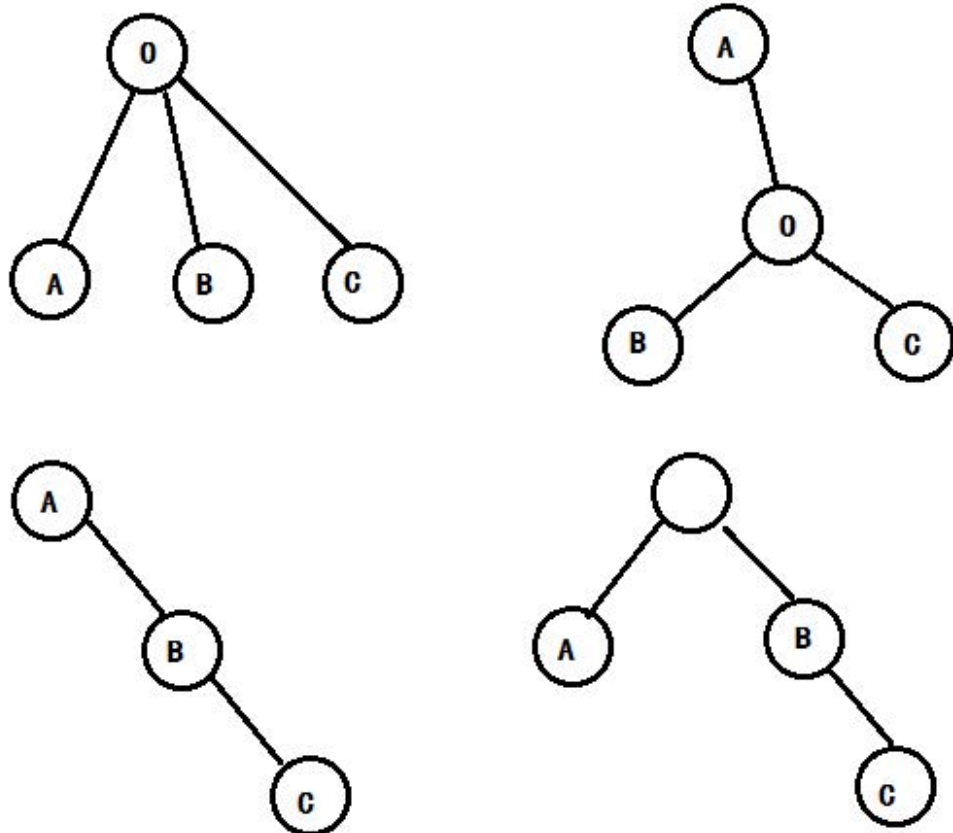


- A-城市轰炸
  - tag: 小模拟 (按题意写程序 ?)
  - 答案是对于每个位置, 输出大于该位置的数的个数 +1。
  - 由于  $n$  不大, 因此可以对于每一个位置都去遍历整个数组, 统计有几个位置的数比它大, 时间复杂度是  $O(n^2)$  的。
- B-收集拼图
  - tag: 贪心
  - 对于每个拼图, 以它为圆心作圆, 求在  $x$  轴上的交线。这样可以得到  $n$  条线段。
  - 那么问题就转化为了, 给出数轴上若干线段, 需要在数轴上选出若干整点, 使得每个线段都至少包含一个选中的整点。
  - 把所有线段对右端点排序。排序后第一个线段的右端点处必选。之后若出现一个线段的左端点大于这个选中的右端点。就将该线段的右端点选中, 重复上述过程。
  - 贪心正确性证明从略。
- C-大可爱大战大恶魔
  - tag: 大模拟
  - 按题意写程序, 参考提供的数据和标程吧
- D-鱼老师爱区间
  - tag: 双指针 (+数据结构优化)
  - 这题建议所有部分分做法都尝试理解实践一下 (
  - 对于 20% 的数据 ( $n \leq 100$ )
    - 对每个询问枚举所有  $\frac{n(n+1)}{2}$  个区间, 对每个区间枚举所有元素统计最大值和最小值, 若满足差值大于  $k$  则答案加 1, 这样做的时间复杂度为  $O(n^3m)$ 。
  - 对于 30% 的数据 ( $n \leq 1000$ )
    - 在上面的基础上加个小优化, 利用  $[l, r]$  的最大最小值和  $a[r+1]$  计算  $[l, r+1]$  的最大最小值可以在枚举所有区间的同时判断当前区间是否满足差值大于  $k$ , 这样做的时间复杂度为  $O(n^2m)$ 。
  - 对于 50% 的数据 ( $n \leq 5000$ )
    - 可以  $O(n^2)$  处理出所有区间的最大值最小值的差值, 把这些值存在一个大小为  $n^2$  的数组里面, 对数组进行排序  $O(n^2 \log^2 n)$ 。对每个询问  $k$  在数组上进行二分, 查找数组中满足大于  $k$  的最小数据位置, 从而得到数组中有多少个差值满足要求, 从而得到询问  $k$  对应的区间个数答案, 总的时间复杂度为  $O(n^2 + n^2 \log^2 n + m \log^2 n)$ 。
  - 对于另外的 20% 的数据 ( $k = 0$ )
    - 显然这个是询问是在所有  $\frac{n(n+1)}{2}$  个区间中有多少区间出现了多于一种数字, 可以用  $\frac{n(n+1)}{2}$  减去只有一种数字的区间个数, 后者可以在遍历数组的同时处理, 即  $O(n)$  计算, 故总的时间复杂度为  $O(n)$ 。
  - 正解: 提供两种做法
    - ST 表 + 双指针: 用  $O(n \log n)$  的时间复杂度预处理出最大值和最小值的 ST 表, ST 表  $O(1)$  查询区间结果 (由于 ST 表查询使用  $\log 2$  函数常数较大, 建议预处理), 利用 ST 表的查询, 对于每个询问, 双指针扫一遍序列得出有多少个区间满足最小值最大值差值大于  $k$ , 总的时间复杂度为  $O(n \log n + mn)$ 。
    - 单调队列 + 双指针: 对每个询问, 双指针扫一遍序列计算有多少个区间满足最小值最大值差值大于  $k$ , 单调队列维护双指针移动过程中区间的最大最小值, 总的时间复杂度为  $O(mn)$ 。
- E-旅游照片
  - tag: 树的性质分析、求LCA
  - 【测试点 1-10】
    - 对于每次询问, 将两条路径上的点全部加上 1 然后统计 2 的数量, 作为答案输出, 询问与询问之间注意清空数组。
    - 该算法的时间复杂度是  $O(nq)$  的。
  - 【测试点 1-16】

- 事实上，我们可以用线段树+树链剖分的方法加速上述过程，注意清空数组时应该 当作为修改改回去，而不能直接 memset，否则复杂度将退化。
- 该算法的时间复杂度是  $O(n + q \log^2 n)$  的。
- 【测试点 11-13、 17-18】
  - 这些测试点满足特殊性质 1，树退化成一条链。
  - 那么，我们只需要判断两个闭区间交的长度即可。
  - 该算法的时间复杂度是  $O(n + q)$  的。
- 【测试点 11、 14-15、 17、 19】
  - 这些测试点满足特殊性质 2。
  - 问题简化成求  $a$  与  $b$  之间的距离，显然可以用树上倍增或 Tarjan 算法求出  $a$  与  $b$  的 LCA，那么
  - $a$  与  $b$  之间的距离就等于  $a$  的深度加  $b$  的深度减去 LCA 的深度的两倍。
  - 该算法的时间复杂度是  $O(n + q)$  或  $O((n + q) \log n)$  的。
- 【测试点 1-20】
  - 注意到问题实际上是问树上一个点到另外两点路径的重合长度，也就是说，这两条路径有一个端点是重合的，不妨就此性质构建算法。
  - 树上三点，或是呈一条链的形式，或是存在一个中心，使得中心到三点之间的路径不相交。如下图，上方的两种情况是  $a, b, c$  点存在中心  $O$ ，下方两种情况是  $a, b, c$  点在一条链上。



- 对于呈一条链的形式，我们不妨将三个点中，位于中间的那个点称为三个点的中心。
- 如此对于四种情况，我们要求的答案都成为了求中心到  $b$  的距离。
- 那么，如何求中心呢？通过观察上图，我们发现，中心一定是  $a, b, c$  点中两点的 LCA。并且，通过进一步观察，我们可以发现，中心是  $a, b, c$  点两两的 LCA 中，深度最大的点。
- 所以，我们只需求出中心，在求出中心到  $b$  点的距离即可解决本题。
- 该算法的时间复杂度是  $O(n + q)$  或  $O((n + q) \log n)$  的。