

基于组合的dfs

比二叉树更难

什么时候使用dfs?

碰到让你找所有方案的题，基本可以确定是dfs

除了二叉树以外90%的dfs题不是排列就是组合

求所有方案可以用bfs嘛? 可以

那用dfs还是bfs? 用bfs

原因: dfs的递归方法好理解，代码短，但是注意递归不一定是好事，当深度非常深的情况，dfs是很浪费时间的，并不是最优解，并且dfs容易stack overflow（深度太深，空间复杂度太大），想解决这个问题就要注意它的递归深度有多深（bfs是不会stack overflow的，原因是bfs是用循环写的不是recursion，stack overflow的先决条件是递归，不递归不会stack overflow。

如果面试官非让用dfs写但是又不想stack overflow

用stack模拟递归就不回stack overflow了，然鹅非常容易写错...

什么是用stack模拟递归? 就是iterator，那道hard的题，BST的iterator，建议背一下BST的interator这道题，其他问题不建议用stack模拟递归，出错率太高，还是怎么简单怎么来（就是用普通递归）

对于排列组合的问题，递归深度会很深嘛?

不会，因为最深就是n了，n又不会很大，有限制

组合的时间复杂度是 2^n

排列的时间复杂度是n!

排列组合类问题一般不需要考虑stack overflow

如果排列组合问题既可以dfs又可以bfs，用bfs/dfs有什么好处?

bfs叫宽度优先搜索，顾名思义，它的空间复杂度取决于图的宽度

dfs叫深度优先搜索，所以空间复杂度取决于图的深度

在搜索类问题中，宽度一般要比深度大，所以dfs的空间耗费是小于bfs的

一般来说，如果面试官不特别要求，dfs都可以用recursion递归的方式来实现

动态规划可以代替dfs嘛?

不能，动态规划只能解决两种问题：

1. 总方案数（一共有多少种方案（返回int值））
2. 方案是否可行

所以在求所有方案本身而不是所有方案的数量时，动态规划dp是无能为力的。

组合搜索类问题

问题模型：求出来所有满足条件的组合

判断条件：组合中的元素是顺序无关的

时间复杂度：与 2^n 相关

递归三要素

递归的定义

递归的拆解

递归的出口