Recursion vs **DFS (Search Algorithm)** **BackTrack (nickName)**

High level

DFS, in more general scope, it is one kind of search algorithm.

DFS can be implemented in an either **recursive way,** or **in iterative Way**

DFS 基本方法:

1. **what does it store on each level?** (每层代表什么意义？一般来讲解题之前就知道DFS要recurse多少层)
2. **How many different states should we try to put on this level?** （每层有多少个状态/case 需要try？）

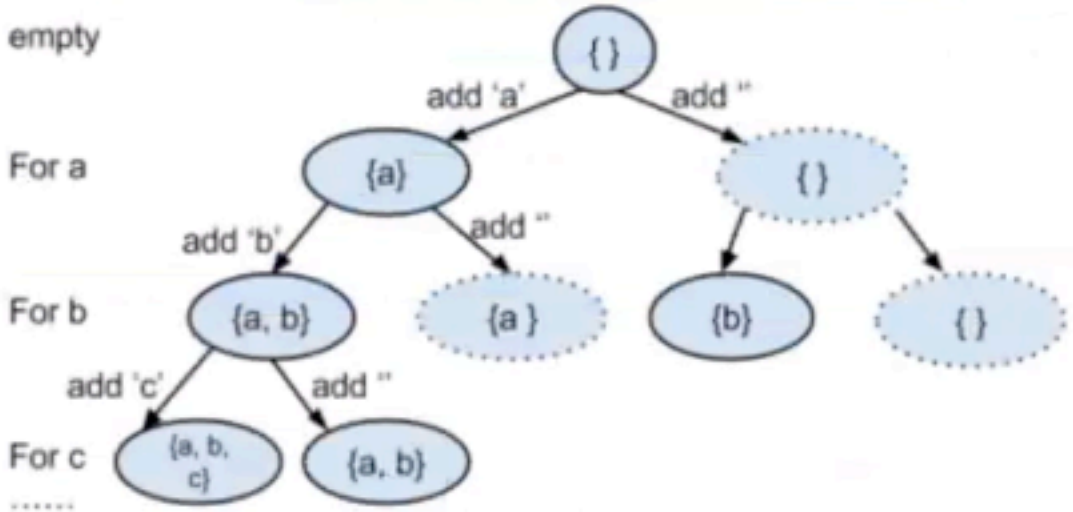
DFS经典例题1 Print all subsets of a set S = {'a', 'b', 'c'}

subset1 = {empty}
subset2 = {a}
....

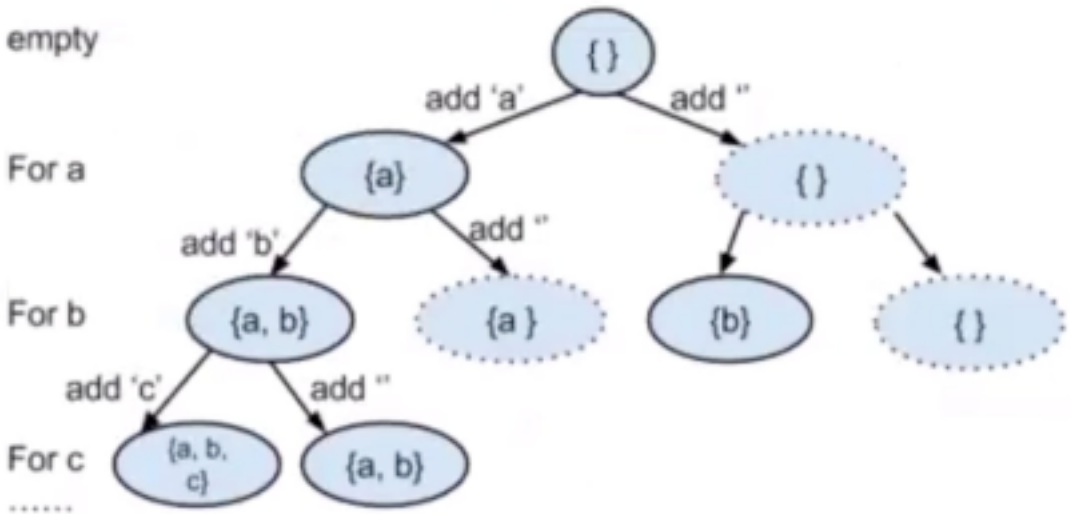
subseti = {a, b, c}

DFS 基本方法:

1. **what does it store on each level? (每层代表什么意义？一般来讲解题之前就知道DFS要recurse多少层)** three levels, each level only consider 1 type of element
2. **How many different states should we try to put on this level? （每层有多少个状态/case 需要try？）** **2 branches (two cases)**

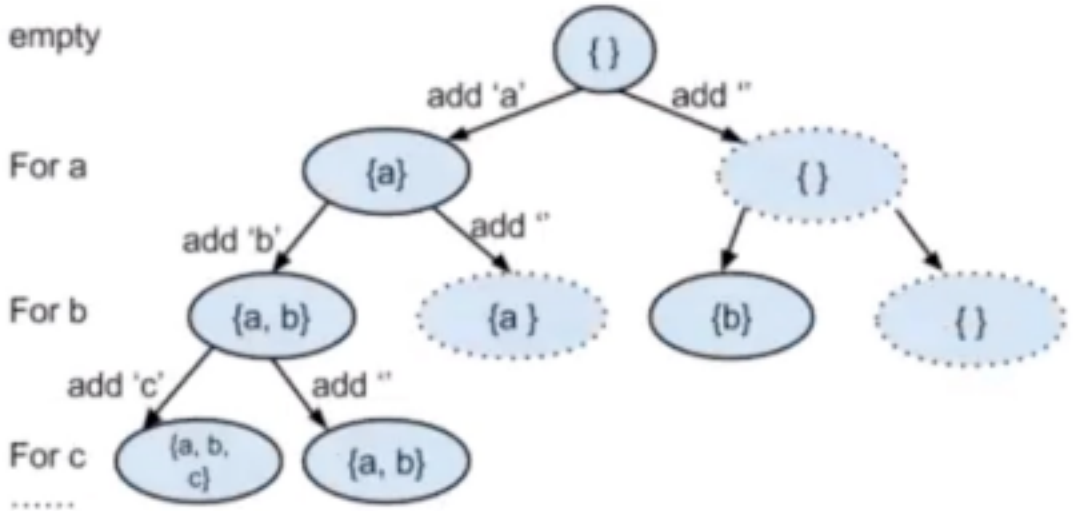


DFS to find all subsets of a set {a, b, c}



DFS to find all subsets of a set {a, b, c}

Time = O(2^n)



DFS to find all subsets of a set {a, b, c}

Time = O(2^n)

Space = O(n)

DFS经典例题3 Print all combinations of coins that can sum up to a total value k.

E.g. total value k = 99 cents

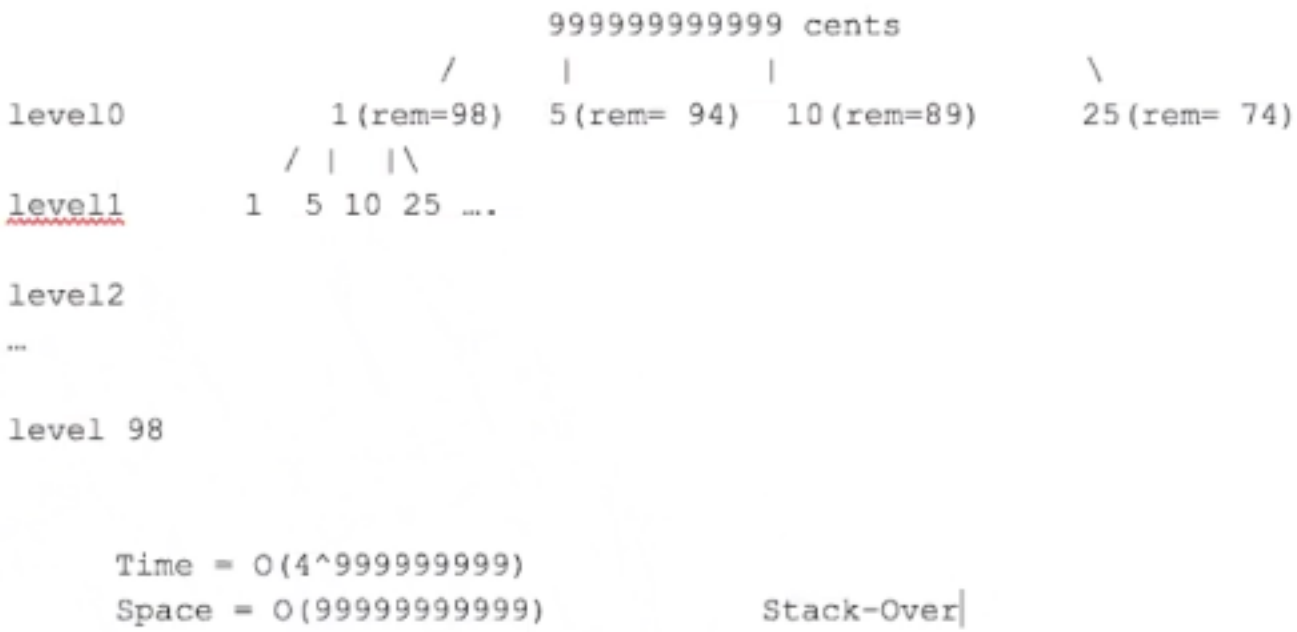
coin value(币值) = 25 10 5 1 cent

Method1: CC189

root (99 cents)

DFS 基本方法:

1. **what does it store on each level? (每层代表什么意义? 一般来讲解题之前就知道DF要recurse多少层)** 99 levels each level represents we take 1 coin out of my hand
2. **How many different states should we try to put on this level? (每层有多少个状态/case 需要try?) 4 branches, each time we can take out 1 kind of coin (1 ,5, 10 ,25)**

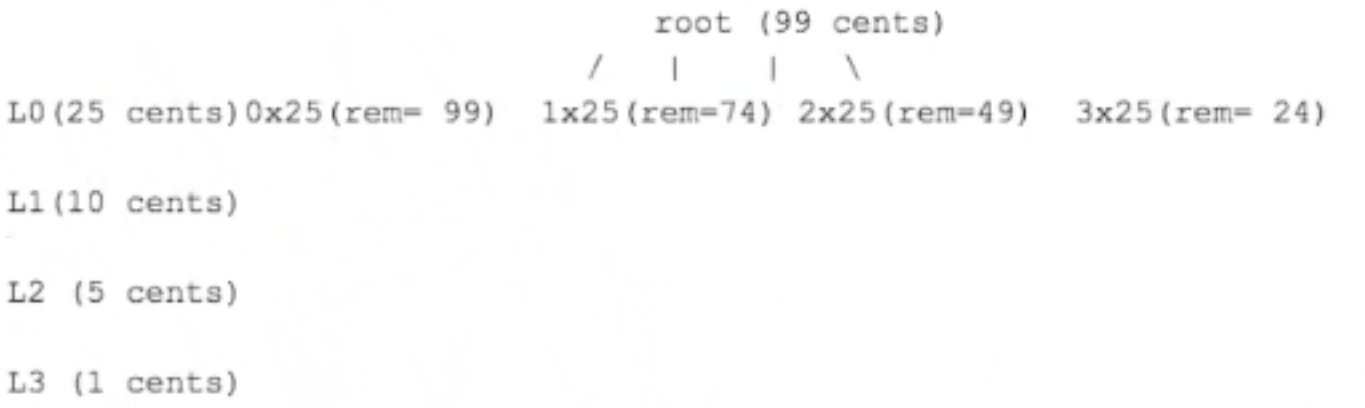

999999999999 cents
level0
1(rem=98)
5(rem= 94)
10(rem=89)
25(rem= 74)
level1
1 5 10 25 ….
level2
…
level 98
Time = O(4^999999999)
Space = O(99999999999)
Stack-Over

DFS 基本方法:

1. **what does it store on each level? (每层代表什么意义? 一般来讲解题之前就知道DF要recurse多少层)** **4 levels** each level represents we try each kind of coins ()

0x25, 1x25 2x25....)

2. **How many different states should we try to put on this level? (每层有多少个状态/case 需要try?)** **dynamic value, depending on the input value**



Conclusion: whenever every single permutation contains all elements in the initial input, then you should consider SWAP and SWAP.

DISCUSSIONS

1. BFS1 vs BFS2
2. BFS1 vs DFS When to use one or the other?
3. Can we use BFS for permutation problems, why ?