1. Generate Parentheses
2. Permutation
3. Subsets(부분집합)
4. Letter Combinations Of a Phonenumber

Generate Parentheses

Problem

Input: 3
Output:
[
  "((()))",
  "(()())",
  "(())()",
  "()(())",
  "()()()"
]

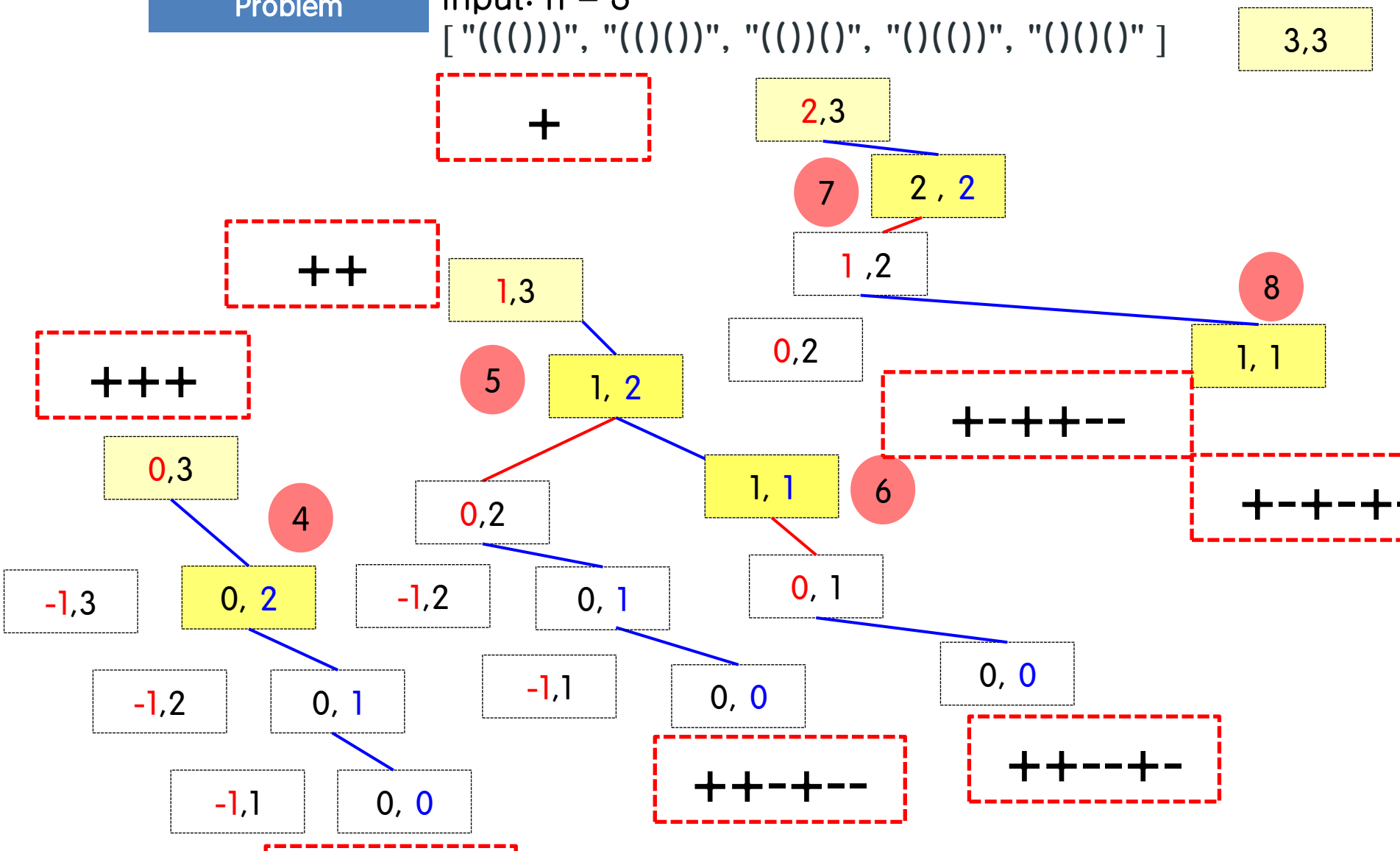
1. Dfs 문제

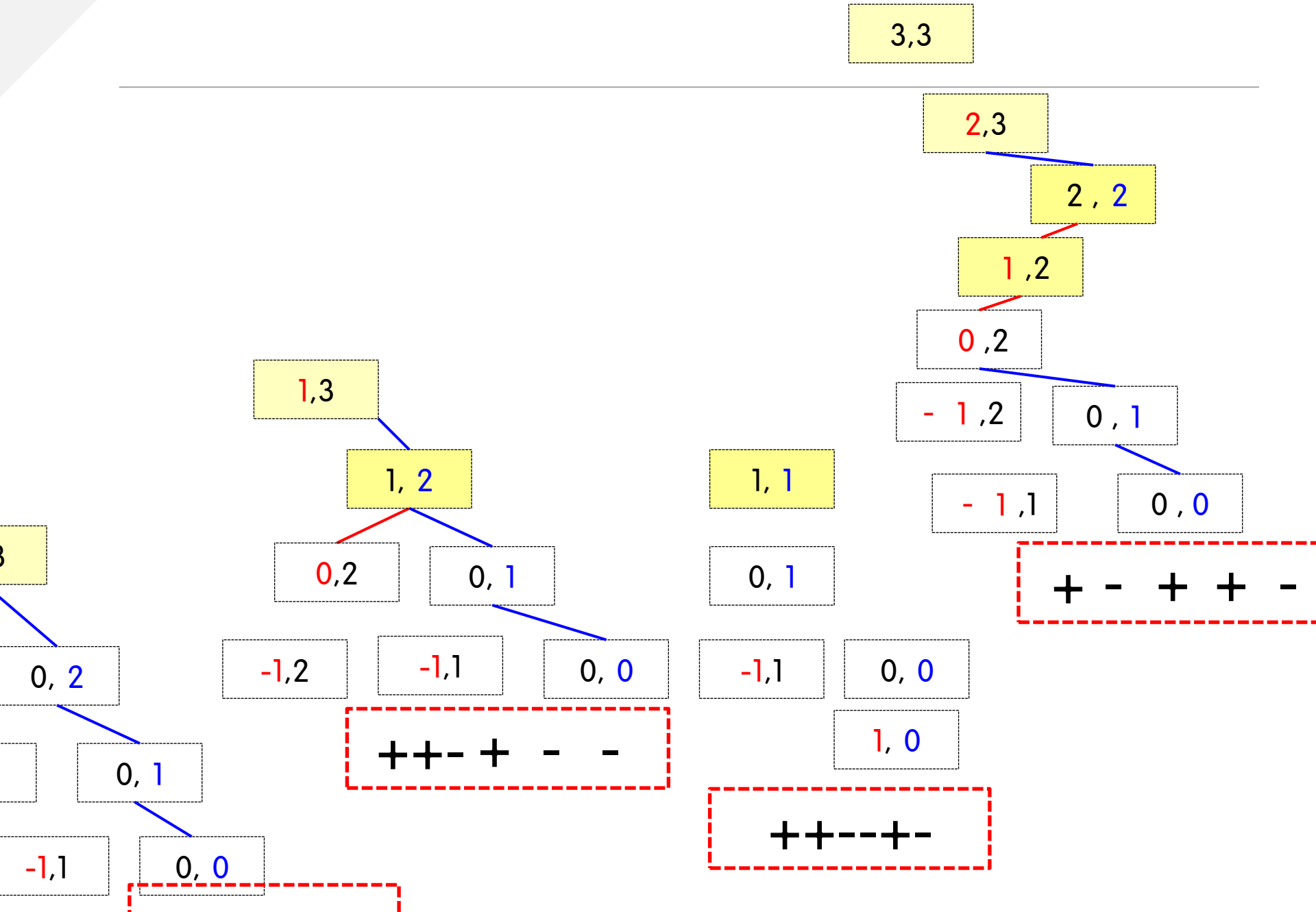
Generated Parentheses

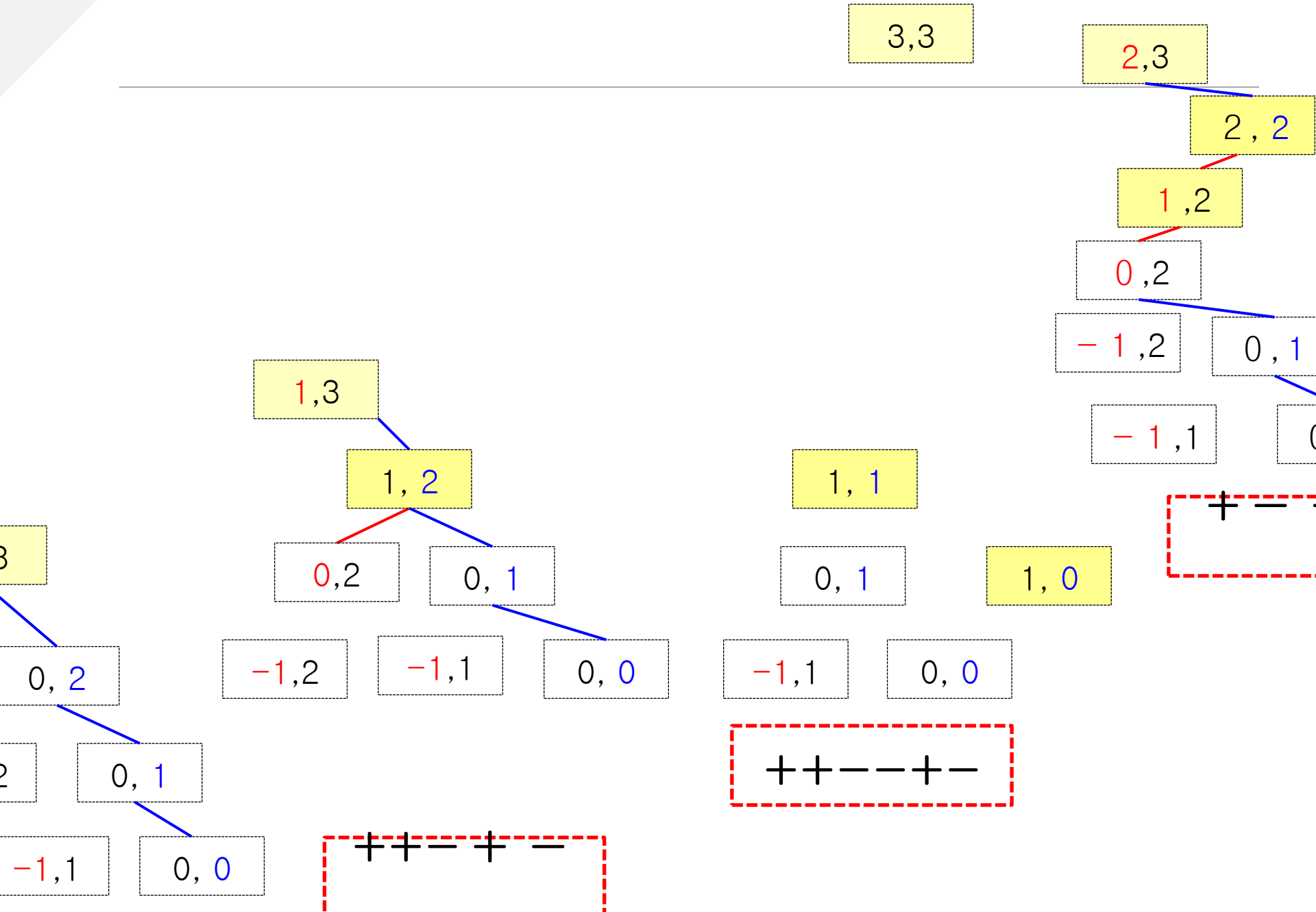
Problem

Input: n = 3

["((()))", "(()())", "(())()", "()(())", "()()()"]




3,3
2,3
2 , 2
1 ,2
0 ,2
- 1 ,2
0 , 1
- 1 ,1
0 , 0
+ - + + -
1,3
1, 2
0,2
0, 1
-1,2
-1,1
0, 0
++- + - -
1, 1
0, 1
-1,1
0, 0
1, 0
++--+-
0, 2
0, 1
-1,1
0, 0

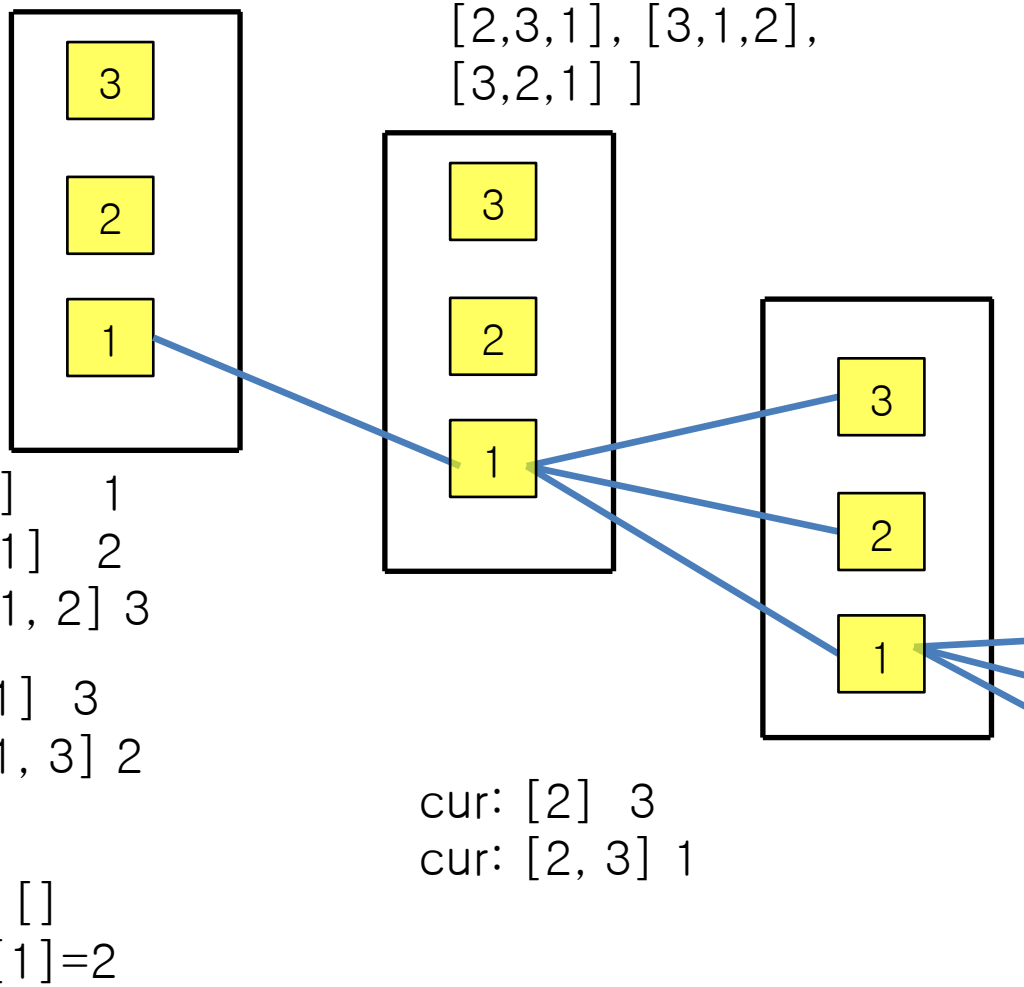

3,3
2,3
2 , 2
1 ,2
0 ,2
− 1 ,2
0 , 1
− 1 ,1
+ −
1,3
1, 2
0,2
0, 1
−1,2
−1,1
0, 0
1, 1
0, 1
−1,1
0, 0
1, 0
++−−+−
0, 2
0, 1
−1,1
0, 0
++−+−

Permutation(퍼뜨리다, 순열)

Problem

Input: [1,2,3]
Output: [[1,2,3], [1,3,2], [2,1,3], [2,3,1], [3,1,2], [3,2,1]]

[1, 1, 1], [1, 1, 2], [1, 1, 3]
[1, 2, 1], [1, 2, 2], [1, 2, 3]
[1, 3, 1], [1, 3, 2], [1, 3, 3]
[2, 1, 1], [2, 1, 2], [2, 1, 3]
[2, 2, 1], [2, 2, 2], [2, 2, 3]
[2, 3, 1], [2, 3, 2], [2, 3, 3]
[3, 1, 1], [3, 1, 2], [3, 1, 3]
[3, 2, 1], [3, 2, 2], [3, 2, 3]
[3, 3, 1], [3, 3, 2], [3, 3, 3]



cur: [] 1
cur: [1] 2
cur: [1, 2] 3

cur: [1] 3
cur: [1, 3] 2

cur: [] 1
nums[1]=2

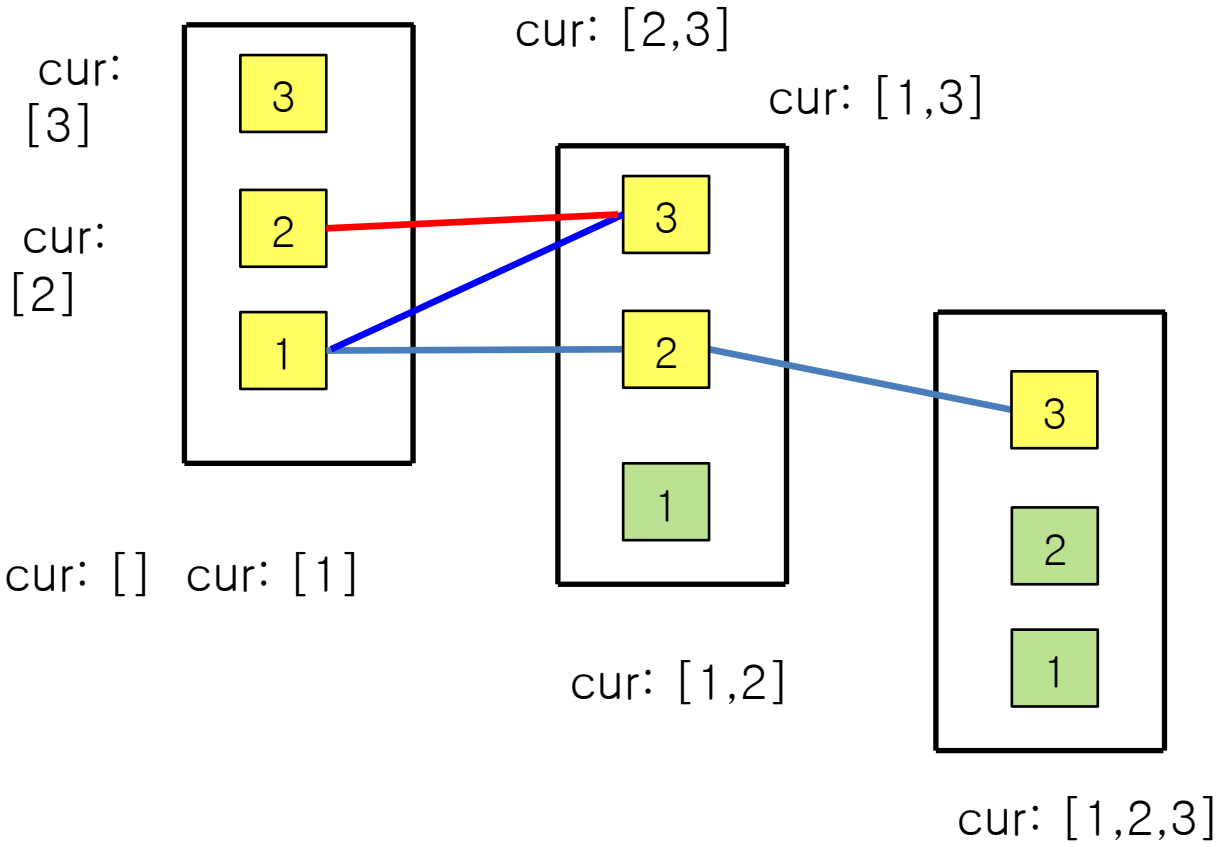
cur: [2] 3
cur: [2, 3] 1

탈출조건 Escape

Subsets(부분집합)

Problem

Input: nums = [1,2,3]
Output:[[3],[1],[2],[1,2,3], [1,3], [2,3], [1,2], []]



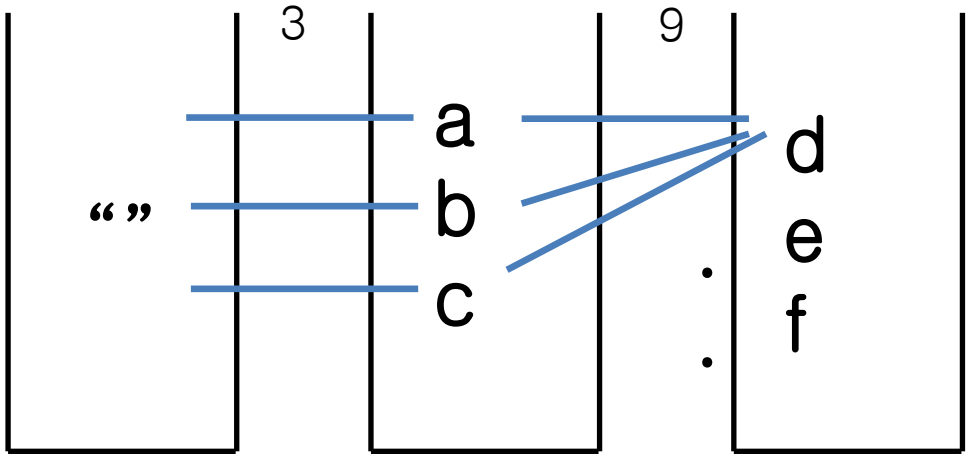
Letter Combinations Of a Phonenumber

String

Problem

Input: "23"
String digitletter[] = { "", "", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", wxyz" };

Output: [ad, bd, cd, ae, be, ce, af, bf, cf]

Example

1. 이중 for 문