# 216. Combination Sum III

## SOLUTION IN JAVA

class Solution {

public List<List<Integer>> combinationSum3(int k, int n) {

List<List<Integer>> ans = new ArrayList<>();

dfs(k, n, 1, new ArrayList<>(), ans);

return ans;

}

private void dfs(int k, int n, int s, List<Integer> path, List<List<Integer>> ans) {

if (k == 0 && n == 0) {

ans.add(new ArrayList<>(path));

return;

}

if (k == 0 || n < 0)

return;

for (int i = s; i <= 9; ++i) {

path.add(i);

dfs(k - 1, n - i, i + 1, path, ans);

path.remove(path.size() - 1);

}

}

}