# 39. Combination Sum

## SOLUTION IN JAVA

class Solution {

public List<List<Integer>> combinationSum(int[] candidates, int target) {

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

Arrays.sort(candidates);

dfs(0, candidates, target, new ArrayList<>(), ans);

return ans;

}

private void dfs(int s, int[] candidates, int target, List<Integer> path,

List<List<Integer>> ans) {

if (target < 0)

return;

if (target == 0) {

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

return;

}

for (int i = s; i < candidates.length; ++i) {

path.add(candidates[i]);

dfs(i, candidates, target - candidates[i], path, ans);

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

}

}

}