Day-18 (Recursion)

Problem-3: Combination sum-1

condi = [2, 3, 6, 7], target = 7

// same no. can be chosen unlimited no. of times.

// For this I used recursion (because same step we do again and again and recursion is a process when f^n call itself).

// So same thing not repeat it automatically done by recursive f^n.

Cases: ① target == 0 // base case.
return;
② target < 0
return

TC: $O(2^{a \cdot target} * k)$

SC: $O(k * x)$

Remarks

// if condi[i] <= tar
v.push_back(condi[i])

// call recursive f^n

// used cand[i], remove from vector using pop f^n.

Problem-9: Combination sum-2

candi = [10, 1, 2, 7, 6, 1, 5]   tar = 8

// candi may only be used once in the combination.

// not contain duplicate.

=> from combi. sum-1 there is only one diff. is not used same ele. again

// if(i!=idx and nums[i]==num[i-1]) continue;

/ call recursi.

/ pop-back();

TC: $O(2^{n*k})$

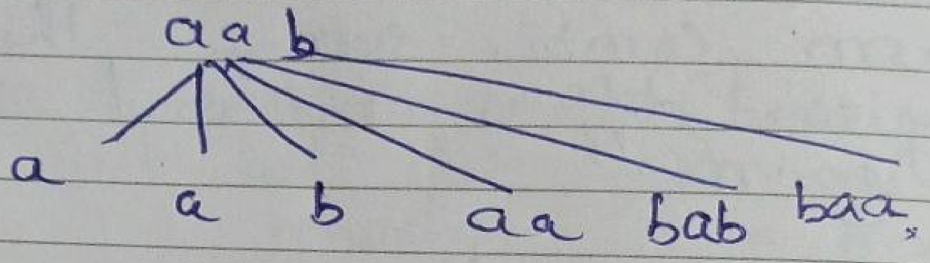
SC: $O(k*x)$

Remarks

Problem-5: Palindrome Partitioning

Input: s="aab"
output: [["a","a","b"],["aa","b"]]

// one thing → a single character is also a palindrome.



So for that →

1. check palindrome
2. recursive fn
3. main fn

using these three got a final answer.

Remarks TC: O((2^n)*k(n/k))
SC: O(2^n)