CLASS 75

03 May 2023 21:10

BackTracking

Combination Sum - I, II, III

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

Subset

2, 22, 222, 2222

Subset ⇒ find all possible self combinations " " " " with relevant Combination Sum II " " " " " Sum target

include
include &

Store all possible combinations

```
vector<vector<int>> combinationSum( _ _ )
{
    vector<int> temp;
    dfs( candidates, target, temp, 0 );
    return ans;
}

vector<vector<int>>ans;
void dfs ( vector<int> candidates, int target, vector<int> temp,
           int start )
{
    if (target == 0)
    {
        ans.push_back (temp);
        return;
    }
    if( start == candidates.size() )
```

```
if( start == candidates.size() )
{ return;
}
```

2x

1 2 3 4

X

```
dfs ( candidates , target , temp , start+1);
```

5

```
if (candidates [start] <= target)
{ temp. Push-back ( candidates [start] );
  dfs ( candidates , target - candidates [start] , temp , start );
}
```

5

6

6 7 8 9 10

temp

3

For loop ko

1 2 3 4

```
For ( int i = start ; i < n ; i++ )
{   if ( candidates [i] <= target)
    { temp. Push-back ( candidates [i] );
      dfs (candidates , target - candidates [i] , temp , i);
      temp .Pop -back();
    }
}
```

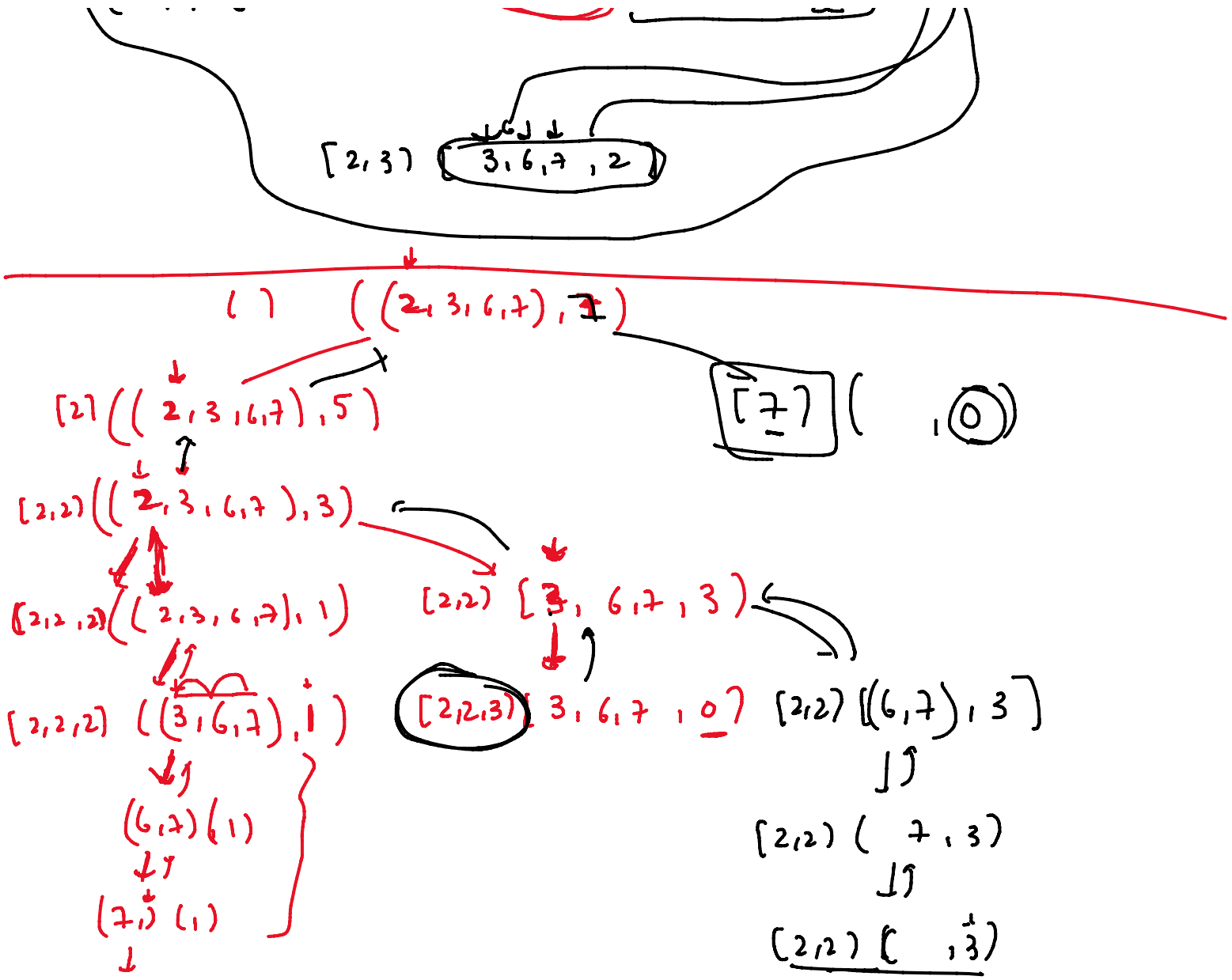
[ ] [2, 3, 6, 7 , 7]

[7] [7 , 0)

[2] [2,3,6,7, 5]

[2,2] [2, 3, 6, 7 , 3]

[2,2,2] [2, 3, 6, 7 , 1) [2,2,3] [3 , 6 , 7 , 0]



Combination sum - II

[10,1,6,7,2,1,5], 8

[1,1,2,5,6,7,10], 8

[1,7]
[7,1]

[1,7]
[1,7]

vector<vector<int>> combinationSum( )

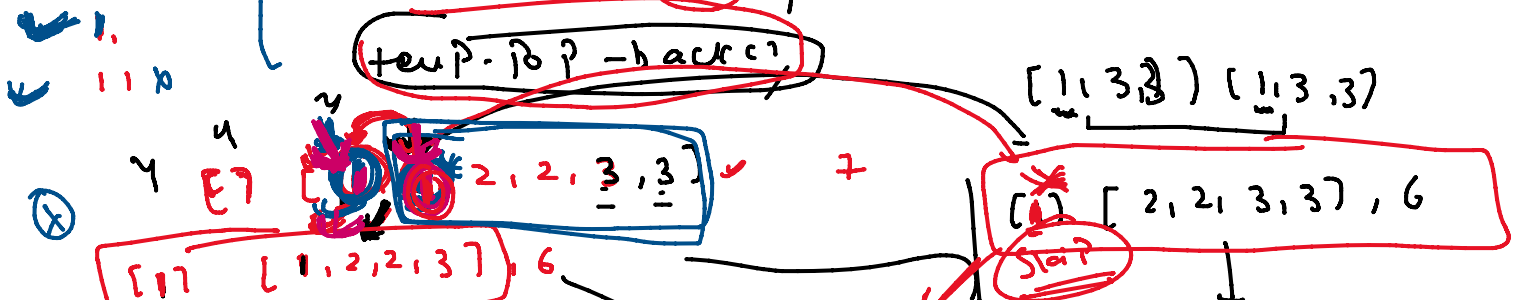
vector<vector<int>> ans;

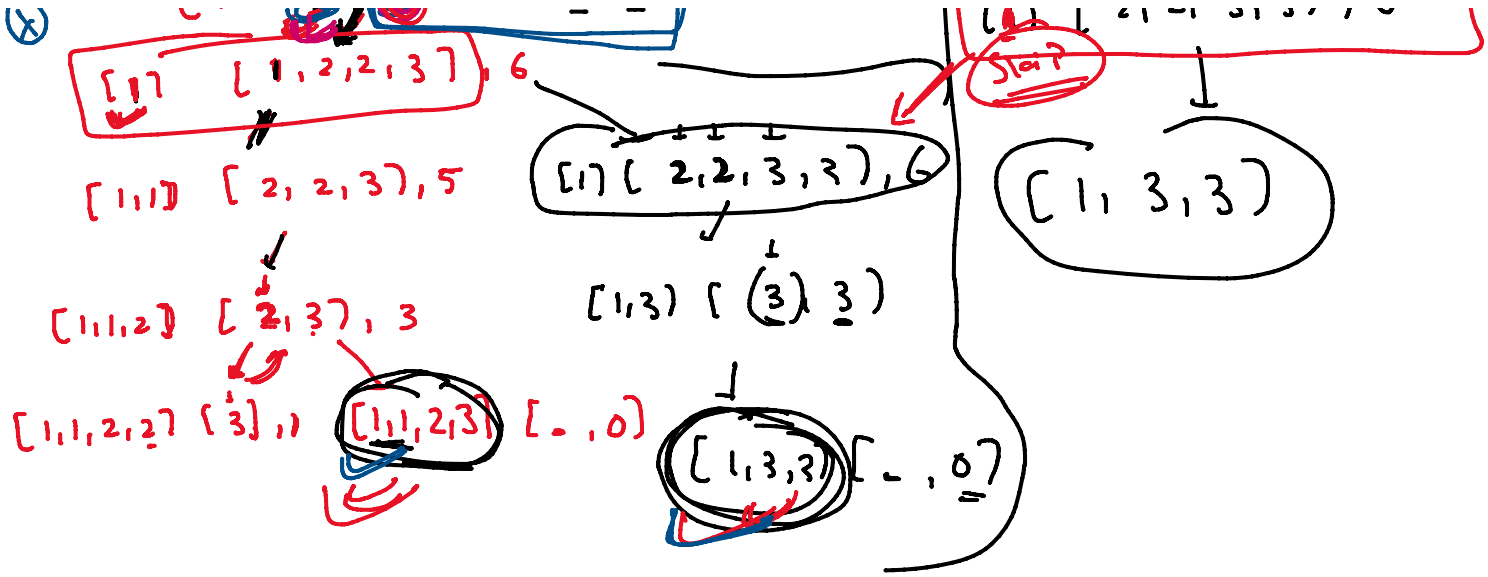
vector<int> temp;

```
vector<int> temp;
sort(candidates.begin(), candidates.end());
solve(candidates, target, ans, temp, 0);
return ans;
}

void solve(vector<int>& candidates, int target, vector<vector<int>>& ans, vector<int>& temp, int idx) {
    int n = candidates.size();
    if (target == 0) {
        ans.push_back(temp);
        return;
    }
    if (idx == n)
        return;
    for (int i = idx; i < n; i++) {
        if (i == idx || candidates[i] != candidates[i-1]) {
            if (candidates[i] <= target) {
                temp.push_back(candidates[i]);
                solve(candidates, target - candidates[i], ans, temp, i+1);
                temp.pop_back();
            }
        }
    }
}
```

Same Base Case





Combination Sum 3

k = 4 , n = 3

[1, 9]

Find all combinations of k size sum upto n

```
vector<vector<int>> ans;
vector<vector<int>> combinationSum3(int k, int n)
{
    vector<int> temp;
    find(k, n, temp, 1);
    return ans;
}

vector<vector<int>> ans;
void find(int k, int n, vector<int> temp, int idx)
{
    if(n==0 and k==0)
    {
        ans.push_back(temp);
        return;
    }
```

(1 - 9)

(1 - 9)

```
    }
    if ( idu == 10 )
    {   return ;
    }
    if(n==0 or k<=0)
        return;
    find ( k, n, temp, idu+1 );
    temp.push_back ( idu );
    find (k-1, n- idu, temp, pau+1);
```

4.