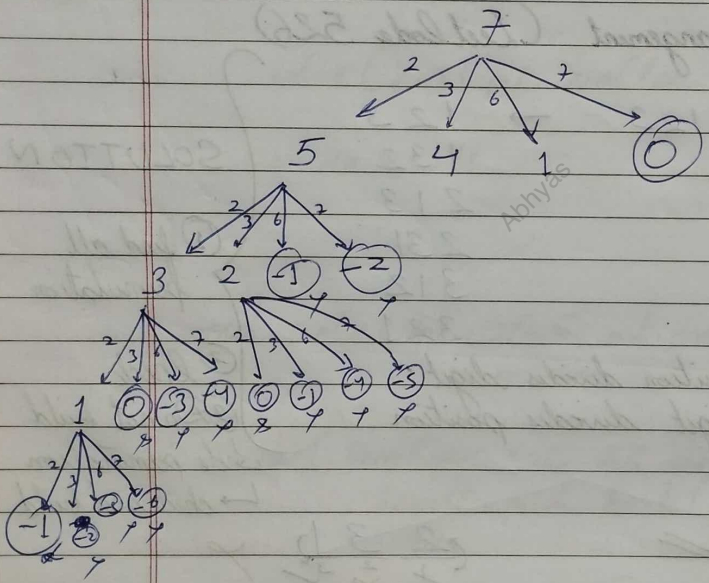
24 Backtracking II

Q ~~Ex~~ Combination Sum

candidate [] = {2, 3, 6, 7}
target = 7

O/P → {[2,2,3], [7]}



BC
if (target == 0)
ans store
return

(target < 0)
return

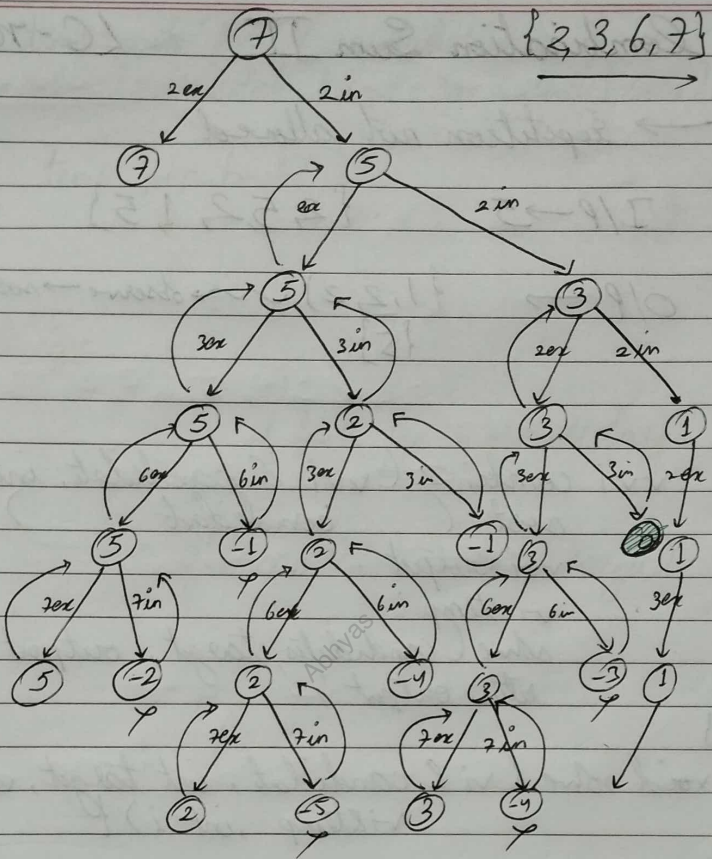
```
vector<vector<int>> combine(vector<int>& candidate,
                            int target) {
    vector<vector<int>> output;
    vector<int> temp;

    solve(candidate, target, output, temp, 0);
    return output;
}
```

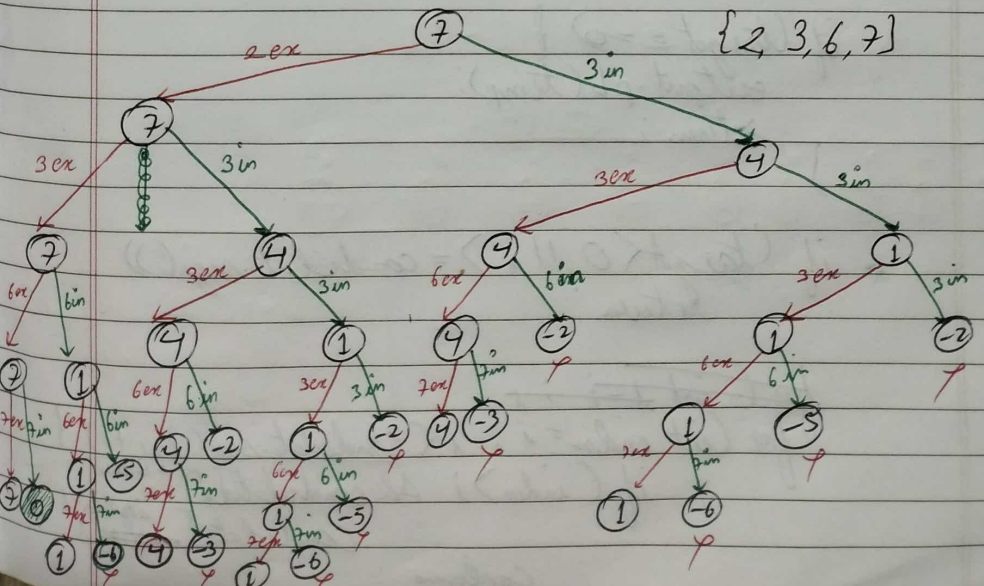
```
void solve(vector<int> & candidate, int target, vector<vector<int>>
                & output, vector<int> & temp, int i) {
    //BC
    if (target == 0) {
        output.push_back(temp);
        return;
    }
    //BC
    if (target < 0 || i >= candidate.size())
        return;

    solve(candidate, target, output, temp, i+1); //exclude

    //include
    temp.push_back(candidate[i]);
    solve(candidate, target - candidate[i], output, temp, i);
    temp.pop_back(); //backtracking
}
```

{2, 3, 6, 7}

sorted

if(candidate[i] > target)


{2, 3, 6, 7}
2ex
2in
esx
3ex
3in
6ex
6in
7ex
7in


{2, 3, 6, 7}
2ex
3in
3ex
6ex
6in
7ex
7in

Q Combination Sum II LC-40

→ repetition not allowed

I/P → 5 {2, 5, 2, 1, 5}

O/P → {1, 2, 2} ⇝ observe → sorted
{5}

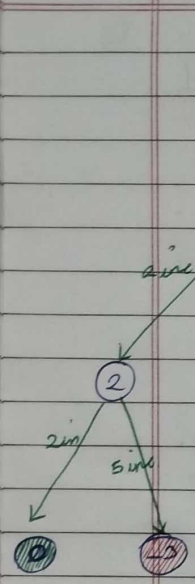
```
vvi combine2 (vvi & candidate, int target)
    sort ( candidate )
    vvi output ;
    vi temp ;
    solve ( candidate, target, output, temp, 0);
    return output ;
}
void solve (vi & candidate, int target, vvi output
        , vi temp , int i) {

    if (target == 0) {
        output.pb(temp);
        return ;
    }

    if (target < 0 || i >= candidate.size())
        return ;

    for (int indx = i ; indx < candidate.size(); indx++) {
        if (indx > i && candidate[indx] == candidate[indx-1])
            continue ;
```

```
temp.push_back(candidates[index]);
solve(candidates, target - candidates[index],
      output, temp, i+1);
temp.pop_back();
}
}
```

~~Optimize~~ {1, 2, 2, 2, 5, 5}

if fails the right all large. don't proceed

```
if(candidates[index] > target)
    break;
```



{2,5,2,1,2}
{1,2,2,2,5}
sort

Q Combination Sum III

k = 3 n = 7

{1,2,3,4,5,6,7,8,9}

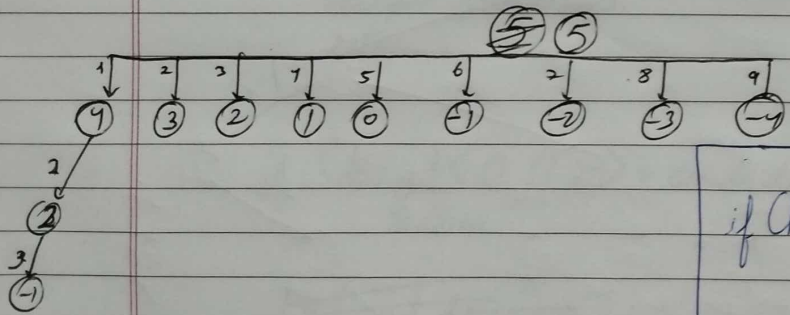
→ no repitition

O/P ⇒ {1,2,4}

I/P k = 3 n = 7

O/P → {1,2,6}, {2,3,4} {1,3,5}

k = 3 n = 5 {1-9}




```
//BC
if (k == 0 && n == 0){
    // ans store
    return
}
```

```
vvi combine(int k, int n)
    vvi output;
    vi temp;

    solve(k, n, temp, 1);
    return output;
}

void solve(int k, int n, vvi &output, vi temp, int num){

    if(k == 0 && n == 0){
        output.push_back(temp);
        return;
    }

    for(int i = num; i <= 9; i++){
        temp.push_back(i);
        solve(k-1, n-i, output, temp, i+1)
        temp.pop_back();
    }
}
```