Date: / /
Page:

Day - 87

Recursion -10 (Subset Sum)

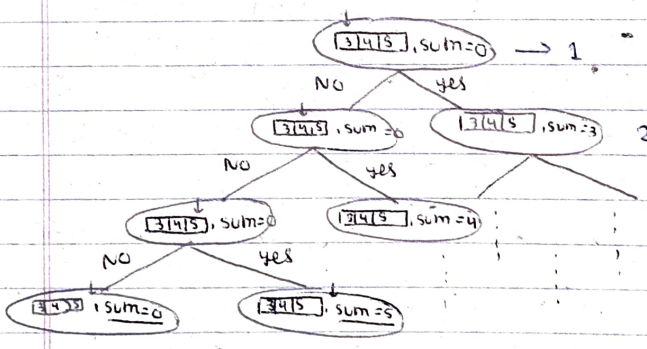
0	1	2
3	4	5

⇒ {3} = 3
{4} = 4
{5} = 5
{3,4} = 7
{3,5} = 8
{4,5} = 9
{3,4,5} = 12
{} = 0

0	1	2
3	4	5

⇒

This 3 have two choices, also 4 & 5. of considering it or not.



→ 1
2
→ 4
→ 8

$[2^0 + 2^1 + 2^2 + \dots + 2^h]$
$= 2^{h+1} - 1 = O(2^n)$
T.C.
S.C = O(n)

⇒ Code

```
void print(int arr[], int index, int n, int sum){
        if(index == n){
            cout << sum << endl;
            return;
        }
        print(arr, index+1, n, sum);
        print(arr, index+1, n, sum+arr[index]);
}
```

Date: / /
Page:

* Target Sum:

⇒

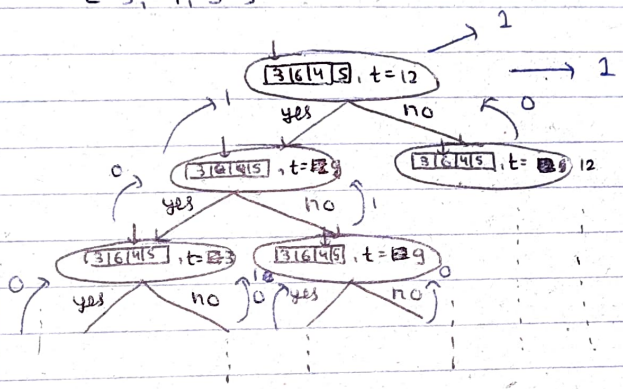
3	6	4	5

Target = 12

⇒ {3,6} X

{6,6} ✓

{3,4,5} ✓



$O(2^n)$ → T.C

Code

⇒

```
bool find( int arr[], int index, int n, int target ){
    if( target == 0 ){
        return 1;
    }
    if ( index == n || target < 0 )
        return 0;
    return find( arr, index+1, n, target ) ||
    find ( arr, index+1, n, target - arr[index] );
}
```

⇒ The biggest advantage of using OR operation is that if it get 1 from anywhere then it knows, the answer will be 1. So, it will not ~~check~~ do remaining fun. calls. This will save our time.