SNS COLLEGE OF ENGINEERING

Coimbatore-107

COURSE NAME: ANALYSIS OF ALGORITHM

II YEAR/ IV SEMESTER

UNIT – V

BACKTRACKING ALGORITHM

Topic

Subset sum Problem

Subset Sum problem

Video: Thadaigalai udaithu..

Given : Using Back Tracking

A set of elements of non negative integer & a value sum.

The Task :

The task is to print the subset of the given set whose sum is equal to the given sum.

eg:

Set = {1, 2, 3, 4}

Sum = 5

Subsets = {1, 4}, {2, 3}

Implementing Backtracking in Subset Sum problem :

Example :

S = {2, 3, 5, 4}

d = 8

The subset can be obtained by State Space search Tree.

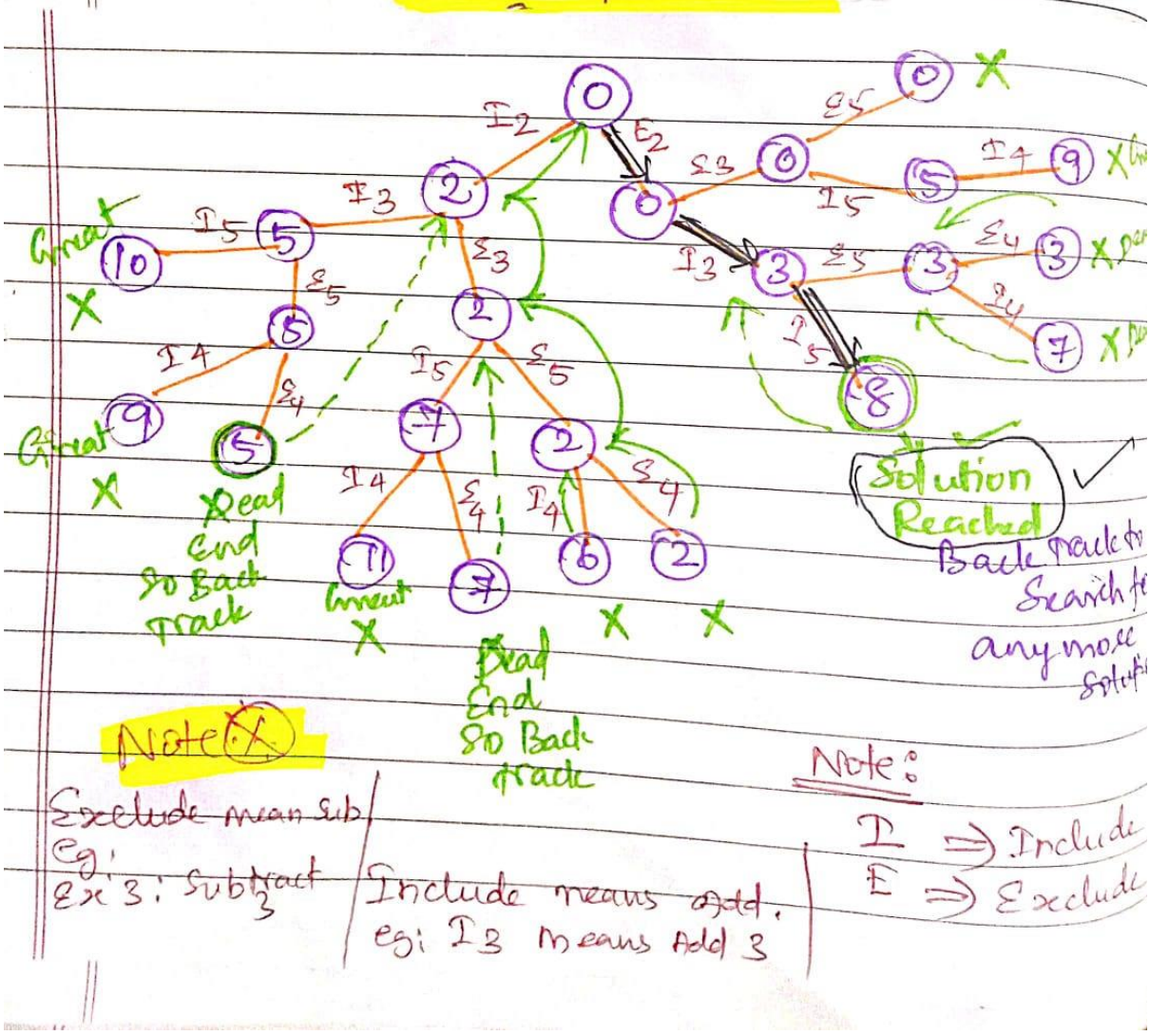
Step 1 : Start from empty set

Step 2 : Include values from set one by one on one side of empty set (on left)

Step 3 : Exclude values from set one by one on one side of empty set (or Right)

⇒ Sum up & include & exclude until all the values are explored
⇒ If there exists no solution on first path, then explore path 2.
⇒ Include & exclude all subset values
⇒ This is done until we reach the solution; i.e. to Sum given.
eg. Sum = 8 in our example.

State Space Tree



Algorithm Subset Sum (set[], n, index, currentSum, target).

```
{
if (currentsum == target)
{ // found valid subset
print subset
return ;
}
if (index == n || currentsum > target)
return;
// include set(index)
include Set[index] in current subset.
Subset Sum ( set , n , index+1 , currentsum
+ set[index], target);
// exclude set [index]
remove set [index] from current subset;
Subset sum (set , n, index+1 , currentSum
target);
```

Time Complexity:

Worst case : $O(2^n)$

Space complexity:

$O(n)$ for recursion Stack (depth atmost n).