AOA EXPERIMENT 10

Name: Mohammed Arhaan Ali
Div: S11
Roll No: 02

Aim: To study and implement the Sum of Subsets Problem.

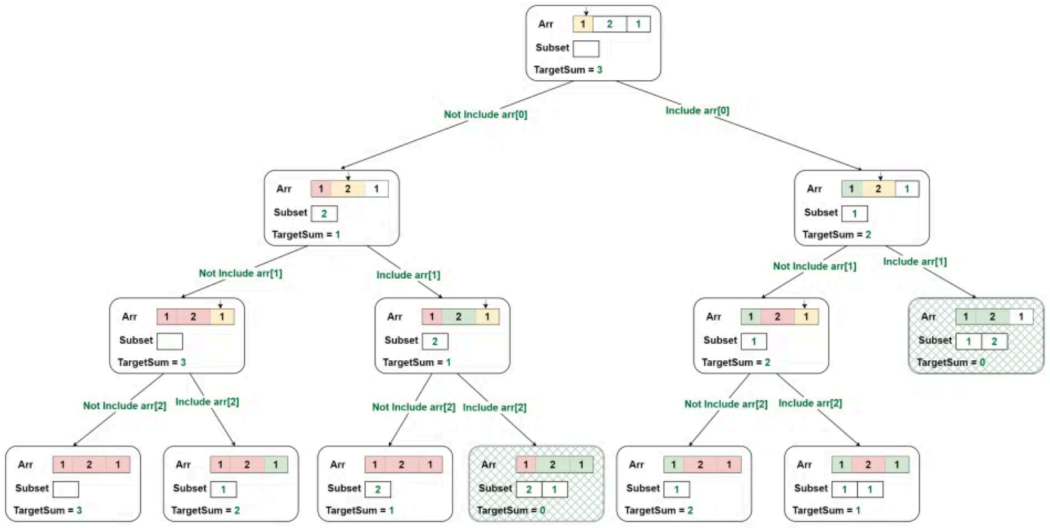
Theory:
Given a set[] of non-negative integers and a value sum, the task is to print the subset of the given set whose sum is equal to the given sum.
Examples:
Input: set[] = {1,2,1}, sum = 3
Output: [1,2],[2,1]
Explanation: There are subsets [1,2],[2,1] with sum 3.
Input: set[] = {3, 34, 4, 12, 5, 2}, sum = 30
Output: []
Explanation: There is no subset that add up to 30.

Subset Sum Problem using Backtracking

Subset sum can also be thought of as a special case of the 0–1 Knapsack problem. For each item, there are two possibilities:

- Include the current element in the subset and recur for the remaining elements with the remaining Sum.
- Exclude the current element from the subset and recur for the remaining elements.

Finally, if Sum becomes 0 then print the elements of current subset. The recursion's base case would be when no items are left, or the sum becomes negative, then simply return.



Subset Sum Problem

Program:

```
#include <stdio.h>
#include <stdlib.h>
static int total_nodes;
void printValues(int A[], int size){
  for (int i = 0; i < size; i++) {
    printf("%*d", 5, A[i]);
  }
  printf("\n");
}
void subset_sum(int s[], int t[], int s_size, int t_size, int sum, int ite, int
const target_sum){
  total_nodes++;
  if (target_sum == sum) {
    printValues(t, t_size);
    subset_sum(s, t, s_size, t_size - 1, sum - s[ite], ite + 1, target_sum);
    return;
  }
  else {
    for (int i = ite; i < s_size; i++) {
      t[t_size] = s[i];
      subset_sum(s, t, s_size, t_size + 1, sum + s[i], i + 1, target_sum);
    }
  }
}
```

```
void generateSubsets(int s[], int size, int target_sum){
   int* tuplet_vector = (int*)malloc(size * sizeof(int));
   subset_sum(s, tuplet_vector, size, 0, 0, 0, target_sum);
   free(tuplet_vector);
}
int main(){
   int set[] = { 5, 6, 12 , 54, 2 , 20 , 15 };
   int size = sizeof(set) / sizeof(set[0]);
   printf("The set is ");
   printValues(set , size);
   generateSubsets(set, size, 25);
   printf("Total Nodes generated %d\n", total_nodes);
   return 0;
}
```

Output:

```
PS C:\Users\arhaa\OneDrive\Desktop\AOA> cd "c:\Users\arhaa\OneDrive\
 if ($?) { .\sumofsubset }
The set is     5    6   12   54    2   20   15
    5    6   12    2
    5   20
Total Nodes generated 127
PS C:\Users\arhaa\OneDrive\Desktop\AOA>
```

Conclusion: Thus we have successfully implemented the Sum of subsets problem.