**Experiment No: 9**

**SUM OF SUBSET PROBLEM**

**Aim:** To implement the sum of subset problem using backtracking

**Theory:**

Backtracking:

The Backtracking is an algorithmic-method to solve a problem with an additional way. It uses a recursive approach to explain the problems. We can say that the backtracking is needed to find all possible combination to solve an optimization problem.

Backtracking is a depth-first search with any bounding function. All solution using backtracking is needed to satisfy a complex set of constraints. The constraints may be explicit or implicit.

Explicit Constraint is ruled, which restrict each vector element to be chosen from the given set.

Implicit Constraint is ruled, which determine which each of the tuples in the solution space, actually satisfy the criterion function.

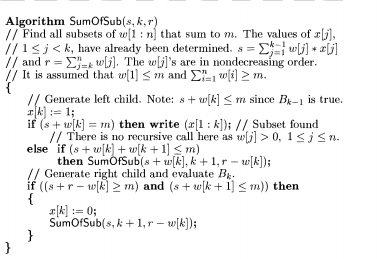
Sum of subset problem:

The Subset-Sum Problem is to find a subset's' of the given set S = (S1 S2 S3...Sn) where the elements of the set S are n positive integers in such a manner that s'∈S and sum of the elements of subset's' is equal to some positive integer 'X.'

For example, consider the list of nums = [1, 2, 3, 4]. If the target = 7, there are two subsets that achieve this sum: {3, 4} and {1, 2, 4}. If target = 11, there are no solutions

The Subset-Sum Problem can be solved by using the backtracking approach. We assume that the elements of the given set are arranged in increasing order:

**Algorithm:**



Conclusion: The sum of subset problem using backtracking was studied and implemented successfully.