



**Bahria University**  
Discovering Knowledge

**BAHRIA UNIVERSITY,**  
**(Karachi Campus)**  
*Department of Software Engineering*  
**Assignment #02– Spring 2022**

COURSE TITLE: **D&AA**  
Class: **BSE 4**  
Course Instructor: **ENGR. BUSHRA FAZAL KHAN**  
Max. Marks: **4 Points**

COURSE CODE: **CSC-321**  
Shift: **Morning**  
Assignment Date: **24-May-2022**  
Assignment Due: **31-May-2022**

Pseudo-code for backtracking algorithm of Sum of subset problem is given below. Explain the mechanism for given data

$n=5$ ,  $W=21$ , and  
 $w_1=5$ ,  $w_2=6$ ,  $w_3=10$ ,  $w_4=11$ ,  $w_5=16$

► Algorithm 5.4

The Backtracking Algorithm for the Sum-of-Subsets Problem

Problem: Given  $n$  positive integers (weights) and a positive integer  $W$ , determine all combinations of the integers that sum to  $W$ .

Inputs: positive integer  $n$ , sorted (nondecreasing order) array of positive integers  $w$  indexed from 1 to  $n$ , and a positive integer  $W$ .

Outputs: all combinations of the integers that sum to  $W$ .

```
void sum_of_subsets (index i,
                    int weight; int total)
{
    if (promising(i))
        if (weight == W)
            cout << include[1] through include[i];
        else{
            include[i + 1] = "yes";           // Include  $w[i + 1]$ .
            sum_of_subsets(i + 1, weight + w[i + 1], total - w[i + 1]);
            include[i + 1] = "no";           // Do not include  $w[i + 1]$ .
            sum_of_subsets(i + 1, weight, total - w[i + 1]);
        }
}

bool promising (index i);
{
    return (weight + total >= W) && (weight == W || weight + w[i + 1] <= W);
}
```