

## Experiment No. 4

Aim: Implement Sum of Subset Problem using Backtracking

Theoretical Background:

- Find a subset of a given set  $S = \{s_1, s_2, \dots, s_n\}$  of  $n$  positive integers whose sum is equal to a given positive integer  $d$ .
- For example, if  $S = \{1, 2, 5, 6, 8\}$  and  $d = 9$  there are two solutions  $\{1, 2, 6\}$  and  $\{1, 8\}$ .
- A suitable message is to be displayed if the given problem instance doesn't have a solution.

Program:

```
#include<stdio.h>

#include<conio.h>

#define TRUE 1
#define FALSE 0

int inc[50],w[50],sum,n;

void sumset(int ,int ,int);

int promising(int i,int wt,int total) {
    return (((wt+total)>=sum)&&((wt==sum) || (wt+w[i+1]<=sum)));
}

void main ()
{
    int i,j,n,temp,total=0;

    clrscr();

    printf("\n Enter how many numbers: ");

    scanf("%d",&n);

    printf("\n Enter %d numbers : ",n);

    for (i=0;i<n;i++) {
        scanf("%d",&w[i]);
        total+=w[i];
    }

    printf("\n Input the sum value to create sub set: ");

    scanf("%d",&sum);

    for (i=0;i<=n;i++)
        for (j=0;j<n-1;j++)
```

```

        if(w[j]>w[j+1]) {
            temp=w[j];
            w[j]=w[j+1];
            w[j+1]=temp;
        }
printf("\n The given %d numbers in ascending order: ",n);
for (i=0;i<n;i++)
    printf("%3d",w[i]);
if((total<sum))
printf("\n Subset construction is not possible");
else{
for (i=0;i<n;i++)
    inc[i]=0;
printf("\n The solution using backtracking is:\n");
sumset(-1,0,total);
}
getch();
}

void sumset(int i,int wt,int total){
    int j;
    if(promising(i,wt,total)) {
        if(wt==sum){
            printf("\n{");
            for (j=0;j<=i;j++)
                if(inc[j])
                    printf("%3d",w[j]);
            printf(" }\n");
        } else
        {
            inc[i+1]=TRUE;
            sumset(i+1,wt+w[i+1],total-w[i+1]);
            inc[i+1]=FALSE;
            sumset(i+1,wt,total-w[i+1]);
        }
    }
}

```

Output:

Conclusion:

Thus, in this experiment we have studied about Sum of Subset problem and how to solve it by using Backtracking algorithm.