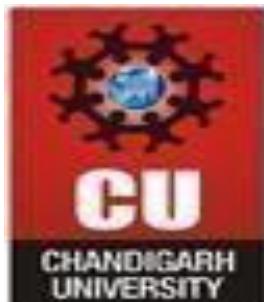




DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

NAME – RAJDEEP JAISWAL

UID – 20BCS2761

BRANCH – CSE BTECH

SEC – WM 902 B

Worksheet Experiment – 2.2

Name: Anant Kumar Mathur

UID: 20BET1071

Branch: BE-IT

Section/Group: 20BET_WM-601-B

Semester: 5th

Subject: DAA Lab

1. Aim/Overview of the practical:

To implement subset-sum problem using Dynamic Programming .

2. Task to be done/ Which logistics used:

find whether or not there exists any subset of the given set .

3. Algorithm/Flowchart:

- i. We create a boolean subset[][] and fill it in bottom up manner.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- ii. The value of subset[i][j] will be true if there is a subset of set[0..j-1] with sum equal to i., otherwise false.
- iii. subset[i][j] = true if there is a subset with:
- iv. the i-th element as the last element * sum equal to j
- v. subset[i][0] = true as sum of {} = 0 vi. subset[0][j] = false as with no elements we can get no sum
- vii. subset[i][j] = subset[i-1][j-E1]; where E1 = array[i-1] viii. Finally, we return subset[n][sum].

4. Steps for experiment/practical/Code:

```
#include<iostream>
using namespace std;

bool subsetsum_DP(int a[],int n, int sum)
{
    bool dp[n+1][sum+1];
    int i,j;

    for(i=0;i<=n;i++)
        dp[i][0]=true;

    for(j=1;j<=sum;j++)
        dp[0][j]=false;

    for(i=1;i<=n;i++)
    {
        for(j=1;j<=sum;j++)
        {
            if(dp[i-1][j]==true)
                dp[i][j]=true;
            else
            {
                if(a[i-1]>j)
                    dp[i][j]=false;
                else
                    dp[i][j]=dp[i-1][j-a[i-1]];
            }
        }
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        }
    }
}
return dp[n][sum];
}

int main() {
    int set[] = { 3, 34, 4, 12, 5, 2 };
    int sum = 9;
    int n = sizeof(set) / sizeof(set[0]);
    if
    (subsetsum_DP(set, n, sum) == true)
        cout <<"Found a subset with given sum";
    else
        cout <<"No subset with given sum";
    return 0;
}
```

5. Observations/Discussions/ Complexity Analysis:

- Worst case time complexity: $\Theta(n^*sum)$
- Space complexity: $\Theta(sum)$

6. Result/Output/Writing Summary:



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
C:\Users\dell\Downloads\Subset_sum.exe
Found a subset with given sum
-----
Process exited after 11.72 seconds with return value 0
Press any key to continue . . .
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C++ program named `Subset_sum.cpp` which implements a dynamic programming solution for the subset sum problem. The terminal window shows the execution results, indicating that a subset was found with the given sum, and the process exited after 11.72 seconds. The status bar at the bottom provides compilation details and system information.

```
File Edit Search View Project Execute Tools AStyle Window Help
C:\Users\dell\Downloads\Subset_sum.cpp - [Executing] - Dev-C++ 5.11
Subset_sum.cpp
1 #include<iostream>
2 using namespace std;
3
4 bool subsetsum_DP(int a[],int n, int sum)
5 {
6     bool dp[n+1][sum+1];
7     int i,j;
8
9     for(i=0;i<=n;i++)
10    dp[i][0]=true;
11
12    for(j=1;j<=sum;j++)
13    dp[0][j]=false;
14
15    for(i=1;i<=n;i++)
16    {
17        for(j=1;j<=sum;j++)
18        {
19            if(dp[i-1][j]==true)
20                dp[i][j]=true;
21            else
22            {
23                if(a[i-1]>j)
24                    dp[i][j]=false;
25                else
26                    dp[i][j]=dp[i-1][j-a[i-1]];
27            }
28        }
29    }
30
31    if(dp[n][sum]==true)
32        cout<<"Found a subset with given sum
33        -----<<endl;
34    else
35        cout<<"No subset found with given sum
36        -----<<endl;
37
38    return 0;
39}
```

Output window:

```
C:\Users\dell\Downloads\Subset_sum.exe
Found a subset with given sum
-----
Process exited after 11.72 seconds with return value 0
Press any key to continue . . .
```

Compiler tab:

```
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation Compilation results...
Line: 19 Col: 37 Sel: 0 Lines: 42 Length: 963 Insert Done parsing in 10.875 seconds
```

System tray:

```
13:10 08-10-2022
```

Learning Outcomes:-

1. Create a program keeping in mind the time complexity
2. Create a program keeping in mind the space complexity
3. Steps to make optimal algorithm
4. Learnt about how to implement subset sum problem using dynamic programming.