



## Department of Computer Science and Engineering (Data Science)

### Experiment 6 (Dynamic Programming)

**Name – DIVYESH KHUNT**

**SAP ID:60009210116**

**Aim:** Implementation of coin change problem using dynamic programming.

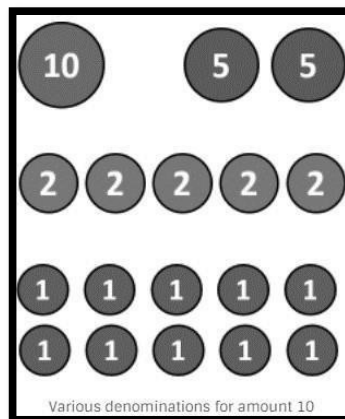
#### Theory:

Making Change problem is to find change for a given amount using a minimum number of coins from a set of denominations.

Explanation: If we are given a set of denominations  $D = \{d_0, d_1, d_2, \dots, d_n\}$  and if we want to change for some amount  $N$ , many combinations are possible. Suppose  $\{d_1, d_2, d_5, d_8\}$ ,  $\{d_0, d_2, d_4\}$ ,  $\{d_0, d_5, d_7\}$  all feasible solutions.

The aim of making a change is to find a solution with a minimum number of coins / denominations. Clearly, this is an optimization problem.

This problem can also be solved by using a greedy algorithm. However, greedy does not ensure the minimum number of denominations.



General assumption is that infinite coins are available for each denomination. We can select any denomination any number of times.

Mathematical Formulation:

$$C[i, j] = \begin{cases} 1 + C[i, j - d_i], & \text{if } i = j \\ C[i - 1, j], & \text{if } j < d_i \\ \min(C[i - 1, j], 1 + C[i, j - d_i]), & \text{otherwise} \end{cases}$$



### **Complexity:**

Best Case Time Complexity:  $O(n)$

### **LabAssignment:**

Write a C Program and consider the set of denominations,  $D=1,4,6$ . Achieve the sum of 8 and calculate the number of coins required and the actual denominations needed using dynamic programming.

### **Code :**

```
9  #include <stdio.h>
10
11
12
13  int min(int a,int b)
14
15  {
16
17      if(a<b) return a;
18
19      else return b;
20
21  }
22
23
24
25  int main()
26
27  {
28      printf("Name: Sachin Nawale \n\n SAP ID: 60009210052\n\n");
29
30      int deno[100];
31
32      printf("enter the number of denominations : ");
33
34      int n=0;
35
36      scanf("%d",&n);
37
38      int i=1;
39
40      while(i<n+1)
41      {
42
43
44          printf("enter the %dth denominations : ",i);
```

```

41
42 {
43
44     printf("enter the %dth denominations : ",i);
45
46     scanf("%d",&deno[i]);
47
48     i++;
49
50 }
51
52 int target=0;
53
54 printf("enter the target : ");
55
56 scanf("%d",&target);
57
58 int dp[100][100]={0};
59
60 for(int j=1;j<target+1;j++)
61
62 {
63
64     for(i=1;i<n+1;i++)
65
66     {
67
68         if((i==1 && i==j) || (i==1))
69
70         {
71
72             dp[i][j]=1+dp[i][j-deno[i]];
73
74         }
75
76         else if(j<deno[i])
77
78         {
79
80             dp[i][j]=dp[i-1][j];
81

```

```

82
83         else if(j<deno[i])
84
85         {
86
87             dp[i][j]=dp[i-1][j];
88
89             dp[i][j]=min(dp[i-1][j],(1+dp[i][j-deno[i]]));
90
91         }
92
93     }
94
95 }
96
97 printf("therefore the least no of coins required is : %d \n",dp[n][target]);
98
99 int solution[100]={0};
100
101 i=n;
102
103 int k=0;
104
105 int j=target;
106
107 while(j>0)
108
109 {
110
111     if(dp[i][j]==dp[i-1][j])
112
113     {
114
115         i--;
116
117         continue;
118

```

```

115         continue;
116     }
117 }
118 else
119 {
120     target=target-deno[i];
121     solution[k]=deno[i];
122     k++;
123     j=target;
124     continue;
125 }
126 }
127 i=0;
128 while(i<k)
129 {
130     printf("the %dth coin in the solution set is : %d \n",i+1,solution[i]);
131     i++;
132 }
133 return 0;
134 }

```

## OUTPUT:

```

enter the number of denominations : 3
enter the 1th denominations : 1
enter the 2th denominations : 4
enter the 3th denominations : 6
enter the target : 8
therefore the least no of coins required is : 2
the 1th coin in the solution set is : 4
the 2th coin in the solution set is : 4

...Program finished with exit code 0
Press ENTER to exit console.

```

**CONCLUSION:** Thus the coin change problem was successfully implemented using dynamic approach.