**Experiment 7**

**Student Name:**Rajiv Paul **UID:**20BCS1812

**Branch:** CSE **Section/Group:** 702 A

**Semester:** 5th **Date of Performance:** 13/09/2022

**Subject Name:** DAA Lab **Subject Code:** 20-CSP-312

**1. Aim/Overview of the practical:**

Code to implement 0-1 Knapsack using Dynamic Programming.

**2. Task to be done/ Which logistics used:**

To write code to implement 0-1 Knapsack using Dynamic Programming.

**3. Algorithm/Flowchart (For programming based labs):**

**4. Steps for experiment/practical/Code:**

package com.DAA;

public class DAA\_exp7 {

public int maxi(int a1, int a2)

{

return Math.*max*(a1, a2);

}

public int maxValueKnapsack(int C, int []w, int[] val, int l)

{

int j, wt;

int [][]dp = new int[l + 1][C + 1];

for (j = 0; j <= l; j++)

{

for (wt = 0; wt <= C; wt++)

{

if (j == 0 || wt == 0)

{

dp[j][wt] = 0;

}

else if (w[j - 1] <= wt)

{

dp[j][wt] = maxi(val[j - 1] + dp[j - 1][wt - w[j - 1]], dp[j - 1][wt]);

}

else

{

dp[j][wt] = dp[j - 1][wt];

}

}

}

return dp[j - 1][C];

}

public static void main(String[] args) {

int []values = new int[] { 10, 6, 8, 7 };

int []weight = new int[] { 7, 9, 3, 8 };

int C = 18;

int l = values.length;

DAA\_exp7 knapObj = new DAA\_exp7();

int maxVal = knapObj.maxValueKnapsack(C, weight, values, l);

System.*out*.println("The maximum value is: " + maxVal);

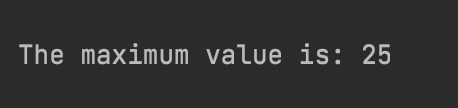
}

}

**5. Observations/Discussions/ Complexity Analysis:**

Time complexity is O(N x sum).

**6. Result/Output/Writing Summary:**

****

**Learning outcomes (What I have learnt):**

**1. Learnt about dynamic programming.**

**2. Learnt how to make optimal algorithm.**

**3. Learnt about 0/1 knapsack problem using dynamic programming.**

**4. Learnt about the implementation of dynamic programming.**

**5. Learnt how to implement 0/1 knapsack problem.**

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |