



Subject :

Software :

Hardware :

Branch :

Semester :

Page No.

Prog No.

PROBLEM STATEMENT

Write a Program to solve Knapsack Problem Using Greedy Algorithm

ALGORITHM &amp; CODE:

```
#include <stdio.h>
int main()
{ float weight[50], profit[50], ratio[50], Total value,
temp, capacity, amount;
int n, i, j;
printf ("Enter the number of items : ");
scanf ("%d", &n);
for ( i=0; i<n; i++)
{
    printf (" Enter profit and weight for item [%d] : \n",
i+1);
    scanf ("%f %f", &profit[i], &weight[i]);
}
printf ("Enter the capacity of knapsack : \n");
scanf ("%f", &capacity);
for ( i=0; i<n; i++)
ratio[i] = profit[i] / weight[i];
for ( i=0; i<n; i++)
for ( j=i+1; j<n; j++)
if (ratio[i] < ratio[j])
```

INPUT GIVEN

INPUT OBTAINED

MARKS



Subject :

Software :

Hardware :

Branch :

Semester :

Page No.

Prog No.

PROBLEM STATEMENT

ALGORITHM &amp; CODE :

```
{   temp = ratio[j];
    ratio[j] = ratio[i];
    ratio[i] = temp;

    temp = weight[j];
    weight[j] = weight[i];
    weight[i] = temp;

    temp = profit[j];
    profit[j] = profit[i];
    profit[i] = temp;
}
```

```
printf ("Knapsack problems using Greedy Algorithm :\n");
for (i=0; i<n; i++)
{ if (weight[i] > capacity)
    break;
else
{ Total value = Total value + profit[i];
  capacity = capacity - weight[i];
}
}
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty

Date :

Signature of Student

Date : 1



Subject :

Software :

Hardware :

Branch :

Semester :

Page No.

Prog No.

PROBLEM STATEMENT

ALGORITHM &amp; CODE :

```
    }
    if(i<n)
        Total value = Total value + (ratio[i] * capacity);
    printf ("The maximum value is : %f\n", Total value);
    return 0;
}
```

### Output obtained

Enter the number of items : 7

Enter Profit and Weight for item [1] : 10 2

Enter Profit and Weight for item [2] : 5 3

Enter Profit and Weight for item [3] : 15 5

Enter Profit and Weight for item [4] : 7 7

Enter Profit and Weight for item [5] : 6 1

Enter Profit and Weight for item [6] : 18 4

Enter Profit and Weight for item [7] : 3 1

Enter the Capacity of knapsack : 15

Knapsack problems using Greedy Algorithm:

The maximum value is : 55.333332

INPUT GIVEN

Time Complexity:  $O(n \log n)$  if Quicksort is used.

OUTPUT OBTAINED

$O(n^2)$  if Bubble Sort is used.

REMARKS

GRADE :

Signature of Faculty

Date :

Signature of Student

Date :