

# **CSE 2003: Lab Assignment #10**

Due on Thursday, April 13, 2017

*Prof. Shaik Naseera 2:00pm*

**Jacob John**

## Contents

Problem 1	3
-----------	---

## Problem 1

Write a program to obtain the solution for fractional knapsack problem using Greedy Method using C

Listing 1: Fractional knapsack problem using Greedy Method using C

```
#include<stdio.h>

void knapsack(int n, float weight[], float profit[], float capacity) {
    float x[20], tp = 0;
    int i, j, u;
    u = capacity;

    for (i = 0; i < n; i++)
        x[i] = 0.0;

    for (i = 0; i < n; i++) {
        if (weight[i] > u)
            break;
        else {
            x[i] = 1.0;
            tp = tp + profit[i];
            u = u - weight[i];
        }
    }

    if (i < n)
        x[i] = u / weight[i];

    tp = tp + (x[i] * profit[i]);

    printf("\nThe result vector is:- \n");
    for (i = 0; i < n; i++)
        printf("%f\t", x[i]);

    printf("\nMaximum profit is:- %f\n", tp);
}

int main() {
    float weight[20], profit[20], capacity;
    int num, i, j;
    float ratio[20], temp;

    printf("\nEnter the no. of objects:- ");
    scanf("%d", &num);

    printf("\nEnter the wts and profits of each object:- ");
    for (i = 0; i < num; i++) {
        scanf("%f %f", &weight[i], &profit[i]);
    }

    printf("\nEnter the capacity of knapsack:- ");
```

```

scanf("%f", &capacity);

50  for (i = 0; i < num; i++) {
    ratio[i] = profit[i] / weight[i];
}

    for (i = 0; i < num; i++) {
55      for (j = i + 1; j < num; j++) {
        if (ratio[i] < ratio[j]) {
            temp = ratio[j];
            ratio[j] = ratio[i];
            ratio[i] = temp;

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

65            temp = profit[j];
            profit[j] = profit[i];
            profit[i] = temp;
        }
    }
70 }

knapsack(num, weight, profit, capacity);
return (0);
}

```

### Output:

```

Jacobs-MacBook-Pro:Downloads jacobjohn$ ./a.out
Enter the no. of objects:- 7
Enter the wts and profits of each object:- 2 10
3 5
5 15
7 7
1 6
4 18
1 3
Enter the capacity of knapsack:- 15
The result vector is:-
1.000000 1.000000 1.000000 1.000000 1.000000 0
.666667 0.000000
Maximum profit is:- 55.333332
Jacobs-MacBook-Pro:Downloads jacobjohn$

```

```

n_queen_backtracking.c  knapsack_greedy.c  +
14  else {
15      x[i] = 1.0;
16      tp = tp + profit[i];
17      u = u - weight[i];
18  }
19  }
20  }
21  if (i < n)
22      x[i] = u / weight[i];
23
24  tp = tp + (x[i] * profit[i]);
25
26  printf("\nThe result vector is:- \n");
27  for (i = 0; i < n; i++)
28      printf("%f\t", x[i]);
29
30  printf("\nMaximum profit is:- %f\n", tp);
31
32  }
33
34  int main() {
35      float weight[20], profit[20], capacity;
36      int num, i, j;
37      float ratio[20], temp;
38
39      printf("\nEnter the no. of objects:- ");
40      scanf("%d", &num);
41
42      printf("\nEnter the wts and profits of each object:- ");
43      for (i = 0; i < num; i++) {
44          scanf("%f %f", &weight[i], &profit[i]);
45      }
46
47      printf("\nEnter the capacity of knapsack:- ");
48      scanf("%f", &capacity);
49
50      for (i = 0; i < num; i++) {
51          ratio[i] = profit[i] / weight[i];
52      }
53
54      for (i = 0; i < num; i++) {
55          for (j = i + 1; j < num; j++) {
56              if (ratio[i] < ratio[j]) {
57                  temp = ratio[j];
58                  ratio[j] = ratio[i];
59                  ratio[i] = temp;
60              }
51          }
52      }
53
54      knapsack(num, weight, profit, capacity);
55      return (0);
56  }
57
58  }
59
60

```