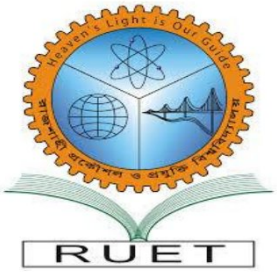
Heaven's Light is Our Guide




# Rajshahi University of Engineering and Technology

## Department of Computer Science and Engineering

**Course No:** CSE.2202

**Course Title:** Sessional based on CSE.2201 (Computer Algorithms)

**Lab Report No:** 04

**Lab Report On:** Greedy Knapsack Algorithm.

**Submitted By**

Md. Ariful Islam

Roll No: 1803046

Section: A

Department: CSE

**Submitted To**

Dr. Md. Ali Hossain

Associate Professor

Dept. of CSE,RUET

**Date: 26-03-2021**

Title: A solution to Greedy knapsack problem for finding the optimal subset of the input data.

Introduction: The greedy method is the most straight-forward design technique and can be applied to a wide variety of problems. Most of the problems will have n inputs and required to find a subset that satisfies some constraints. Any subset that satisfies the constraints is called feasible solution. A feasible solution that either maximizes or minimizes a given objective function is called an optimal solution.

Description: In a knapsack problem we will be given n objects and a knapsack or bag. Object i will have a weight $w_i$ and a profit $p_i$ for $w_i$. The knapsack will have a capacity m. If a

fraction $x_i$, $0 \leq x_i \leq 1$, of object $i$ is placed into the knapsack, the a profit $p_i x_i$ is earned. The objective of knapsack problem is to maximize the profit by filling the knapsack and the total weight of all chosen objects to be at most $m$. Formally, the problem can be stated as,

$$\text{maximum} \sum p_i x_i$$

$$\text{subject to,} \sum w_i x_i \leq m$$

$$\text{where} \quad 0 \leq x_i \leq 1, \quad 1 \leq i \leq n$$

$w_i$ = weight of object $i$

$p_i$ = Profit of object $i$

$x_i$ = fraction of $w_i$.

## Algorithm:

```
Greedy knapsack (m, n) {
// Objects are ordered as  P[i]/w[i] >= P[i+1]/w[i+1]
// m is the size of knapsack
// n is the number of objects.

    for i:= 1 to n do  x[i] = 0.0
    U := m
    for i:= 1 to n do {
        if ( w[i] > U) then break;
        x[i]:= 1.0  ; U = U - w[i];
    }
    if ( i <= n) then  x[i] := U/w[i]

}
```

**Input:** From the "greedy.txt" file.

**Output:** The output will be like this -

"F:\4th Semester\CSE\CSE 2202\Lab 4\greedyKnapsack.exe"

```
Knapsack: 20
Item    Weight    Profit
-------------------------
 1       2        3
 2       4        5
 3       5        8
 4       3        4
 5       9        10

Solution: 1

Fractional Amount: 1 1 1 1 0.666667
Total Weight: 20
Total Profit: 26.6667

Knapsack: 21
Item    Weight    Profit
-------------------------
 1       4        8
 2       6        12
 3       9        4
 4       8        11
 5       5        12
 6       7        4
 7       4        12
 8       8        12

Solution: 2

Fractional Amount: 1 1 0 0 1 0 1 0.25
Total Weight: 21
Total Profit: 47

Knapsack: 26
Item    Weight    Profit
-------------------------
 1       8        3
 2       5        9
 3       3        8
 4       8        6
 5       8        3
 6       7        10
 7       8        10
 8       9        4

Solution: 3

Fractional Amount: 0 1 1 0.375 0 1 1 0
Total Weight: 26
Total Profit: 39.25
```

"F:\4th Semester\CSE\CSE.2202\Lab 4\greedyKnapsack.exe"

```
Knapsack: 22
Item   Weight   Profit
--------------------------
 1      9       11
 2      8       12
 3      3       8
 4      3       12
 5      3       12
 6      6       6
 7      4       8
 8      4       9
 9      9       9

Solution: 4

Fractional Amount: 0 0.625 1 1 1 0 1 1 0
Total Weight: 22
Total Profit: 56.5

Knapsack: 21
Item   Weight   Profit
--------------------------
 1      8       9
 2      4       9
 3      7       8
 4      7       10
 5      4       10

Solution: 5

Fractional Amount: 0 1 0.857143 1 1
Total Weight: 21
Total Profit: 35.8571

Knapsack: 29
Item   Weight   Profit
--------------------------
 1      4       6
 2      7       9
 3      6       3
 4      5       10
 5      4       10
 6      3       9
 7      8       9
 8      5       8

Solution: 6

Fractional Amount: 1 1 0 1 1 1 0.125 1
Total Weight: 29
Total Profit: 53.125
```

"F:\4th Semester\CSE\CSE 2202\Lab 4\greedyKnapsack.exe"

```
 Knapsack: 25
 Item    Weight    Profit
----------------------------
  1       8         6
  2       4         9
  3       4         12
  4       9         7
  5       7         5

 Solution: 7

 Fractional Amount: 1 1 1 1 0
 Total Weight: 25
 Total Profit: 34

 Knapsack: 26
 Item    Weight    Profit
----------------------------
  1       7         12
  2       7         6
  3       7         12
  4       5         9
  5       7         10
  6       4         5
  7       7         12

 Solution: 8

 Fractional Amount: 1 0 1 1 0 0 1
 Total Weight: 26
 Total Profit: 45
```

"F:\4th Semester\CSE\CSE 2202\Lab 4\greedyKnapsack.exe"

```
 Knapsack: 27
 Item   Weight   Profit
--------------------------
  1      8       12
  2      8       10
  3      4       11
  4      8       12
  5      8       6
  6      8       10

 Solution: 9

 Fractional Amount: 1 0.875 1 1 0 0
 Total Weight: 27
 Total Profit: 43.75

 Knapsack: 25
 Item   Weight   Profit
--------------------------
  1      9       9
  2      5       6
  3      6       6
  4      3       8
  5      3       7
  6      4       12
  7      8       6

 Solution: 10

 Fractional Amount: 0.444444 1 1 1 1 1 0
 Total Weight: 25
 Total Profit: 43

Process returned 0 (0x0)   execution time : 0.347 s
Press any key to continue.
```

Discussion and Conclusion: In the problem we had to determine the profit per weight first. Then from the maximum values we started filling the knapsack.

We ~~could~~ had successfully implemented the Greedyknapsack algorithm in the code and get the maximized profit for n sized knapsack.