IE 310-Programming Assignment 1
Due on February 21, 2020 17:00

In this assignment, you are going to write a program which solves Knapsack Problem using Greedy Heuristic. Knapsack Problem is given as follows: Let there be n objects with some profit (value) p_i and volume v_i, i = 1,…,n. Our aim is to maximize the profit by choosing a subset of those objects while satisfying the constraint that the sum of volumes of objects chosen does not exceed the volume of our knapsack B. Then, this problem can be formulated as an integer programming model as follows:

$$\max \sum_{i=1}^{n} p_i x_i$$

$$s.t. \sum_{i=1}^{n} v_i x_i \leq B$$

$$x_i = 0,1 \qquad i = 1, \dots, n$$

Here, x_i is a binary variable taking value one if object i is chosen and zero otherwise.

Note that Knapsack Problem is NP-Complete. Therefore, heuristic algorithms are proposed to solve it (heuristic algorithms are used to solve difficult optimization problems in a reasonable amount of time, but they do not give optimal solutions in general). One such heuristic is called the Greedy Heuristic and its pseudocode is given as below:

```
Sort the objects in descending order of their $ri = pi/vi$ values and rename as $[i]$.
Let $W = 0$
**for** [i] = 1 to n **do**
    **if** $W + v_{[i]} \leq B$ **then**
            put object $[i]$ into the knapsack and $W = W + v_{[i]}$
    **end if**
**end for**
Report the objects chosen and the total profit
```

Figure 1: Greedy Heuristic for Knapsack Problem.

You should write a computer program in C, C++, R or Python which correctly implements the above algorithm. You need to test your algorithm with the three problem instances attached. The first number in .txt files gives the number of objects and the second number is the capacity of the knapsack. Then, there is an $nx2$ matrix given. The row i represents object i whereas columns represent profit and cost of each object, respectively.

Do not change the input format. You need to submit a brief report which contains description of your sorting algorithm, found results of each instances, and brief discussion about the results. Also, include your source code into your .zip folder. Please name your folder as “Name-Surname-ID-Assignment1”. Submit your folders via Moodle page until due time.

Note: Quality of your sorting algorithm is not going to affect your grade as long as it works correctly. Use of any package/built-in function for sorting is not allowed. Please, write your own sorting (any sorting method could be used).