Design and Analysis of Algorithms
CS375 Spring 2020

Programming Assignment 2

Release Date: 4/1/2020
Due: 4/15/2020 (Wednesday) at start of class

In this assignment, you need to implement the best-first-search branch and bound algorithm for the 0-1 knapsack problem introduced in the class. The algorithm needs to find one optimal solution as quickly as possible. Note: the input problem instance may not contain items in decreasing order of profit/weight, so **your program needs to sort the items first before the search**.

The program will have 2 command line arguments: input file, output file.

Input file:
The structure of the input file is as follows.

Line 1: n,c // n is the size of the problem (the total number of items) and C is the total weight limit
Lines 2 to n+1: w_i,p_i // w_i is the weight of item i and p_i is the profit of item i

The example problem from slide 12 of Lecture 25 can be converted into the following input file format:

```
4,16
2,40
5,30
10,50
5,10
```

Output file:
The structure of the output file is as follows.

Line 1: the size of the problem (i.e., the total number of items),the optimal value (i.e., the maximum profit),the size (k) of the first optimal solution found (i.e., the size of the first optimal subset of items found to produce the maximum profit)
Line 2: the total number of nodes visited (by the algorithm from the start to the end of the search including the root and the last node visited in the binary search space tree),the total number of **leaf** nodes visited
Lines 3 to 3+(k-1): each line contains the weight and profit of an item selected in order for the first optimal solution found.

The solution for the above example in the output file format:

```
4,90,2
11,6
2,40
10,50
```

Please run your code with the provided sample input file above (and other test cases developed by yourself), and check the output carefully to ensure that your program is correct. The TA will use additional input files to test the correctness of your program.

Directions and Requirements:

1. Your programs should be run like this.
Prompt> BestFirstSearch <input.txt> <output.txt>

2. Submit a *.zip* (or tar) file through the submission link at Mycourses.
The zip file should be named (lower case) as follows:
<last name>_<first name>
When the file is unzipped it should contain a directory with the same name as the zip file. The directory should contain the following files:

File(s) for the source code and a Makefile.

README file named as *readme.txt*

The makefile should produce an executable called "BestFirstSearch" for C or C++ or a "BestFirstSearch" class for java

Declaration of Academic Integrity file named declaration.txt which **include the following statements and your full name**:

"I, ___________, have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of **0** for the involved assignment for my first offense and that I will receive a grade of **"F" for the course** for any additional offense."

3. Additional general requirements for programming assignments from the course syllabus (**see item 5**) also must be followed.