CS575
Fall 2016

Design and Analysis of Algorithms
Programming Assignment 3

Due: In Blackboard by **Monday 11/28/2016, 11:59 PM**

You need to implement the dynamic programming algorithm for the knapsack problem. You are only interested in getting one optimal solution as quickly as possible.

Input file
The input file will contain a sequence of problems and will have the following lines:
Line 1 will contain: n W
Lines 2 to n+1: w p
If the file has more than 1 problem Line n+2 will have n2 W2 and lines n+3 to n+n2+2 w p.
For example:

First Problem	Second Problem
3 10 5 70 4 4 3 50	**4 20** 3 10 10 10 6 15 18 100

The first knapsack problem has 3 items and W=10. The next problem has 4 items and W=20.

The Output file:
The output file will contain in line 1 the size of the problem n, the maximum profit, and the time in seconds it took to solve the problem using the algorithm. You will also print the items in an optimal solution with each line containing the item's weight, and profit. Note that there may be more than one solution with the same max profit.

Please run your code with the files

1. input.txt. This file has 10 knapsack problems with sizes 20 to 29.
2. badInput1.txt. This file contains only 1 problem with 8 items. It has a very large W=10,000,000. In this case our algorithm will be quite slow. Note Please note the change in computation time when W>>n! = 8! = 40320 (There are other versions of the algorithm where the worst case time complexity is $O(2^n)$, and you do not need a table of size (n+1)*(W+1)).

CS575
Fall 2016

Submission instructions

1. You may write the code using C, C++, or Java. **Your program should compile on Remote CS(remote.cs.binghamton.edu) machine. No exceptions.** If you have trouble accessing the remote server, please contact CS system administrator. It should be purely a command line program. NO GUI will be accepted. If the submission file is incorrect/ incorrect version and you haven't run on **remote cs machines** , we cannot grade your assignment so you will get zero.

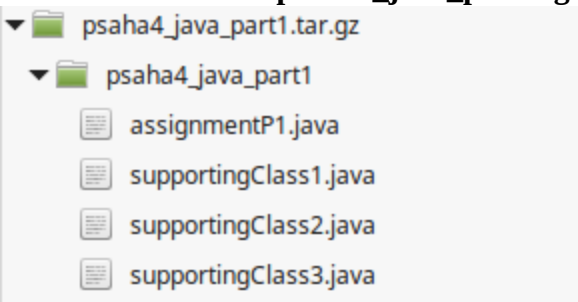
2. Your need to submit a .**tar.gz** file (not only .tar) which should follow the following naming convention. <usrid>_<language>_p3.tar.gz, after unpacking this.tar.gz it should have a directory named <usrid>_<language>_p3. Example: if your user id is psaha4 and you are writing the assignment in java then your submission file name will be **psaha4_java_p3.tar.gz**. For c++ and c developers it will be psaha4_cpp_p3.tar.gz and psaha4_c_p3.tar.gz respectively.

3. Your submission folder should **not** contain input files, object or class files. Presence of these file will cost you 5 points. It should contain a make file, readme file and source code files, not having these files will have 5% penalty.

4. Readme file should contain compilation/running instructions, issues or completed extent of your code (complete or incomplete code) and any further instructions that TA should know about your code. Please note that TA doesn't assume anything about your code and it's your responsibility to mention all required details about your code.

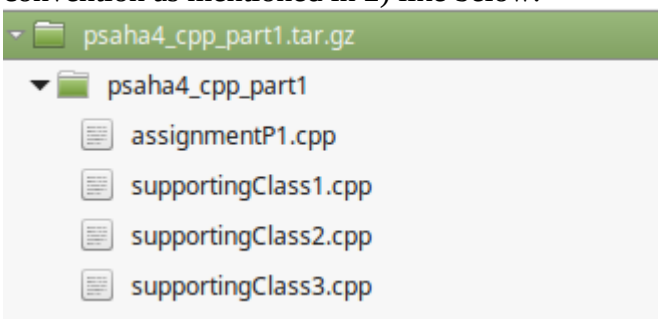
5. Output format: Please follow the format described in the first page.

6. **For Java developers** your entry point class, which contains the main method, should have the name as "asignmentP3.java". So make sure your makefile will produce a assignmentP3.class file after compilation and it can be run by the following command **java assignmentP3 <file path>/<input file name> <output file name>**
7. All your .java files should be inside a single folder, don't create many packages to place your java files. All source code files should be inside a single directory (name should be in this format **psaha4_java_p3.tar.gz**.) like below.



4. **For C++ and C developers** please write your makefile such a way that it produces object file with name assignmentP3.obj and it can be run with the below command **./assignmentP3.out <file path>/<input file name> <output file name>**

All source code files should be inside a single directory (please follow the naming convention as mentioned in 2) like below.



5. Your code should take inputs via input file and output the result via output file. Not implementing file handling operations will have 20% penalty.
6. Hard coding of any filename and file path will cost you 5 points, so make sure your program can take parameters dynamically.
7. Hard coding of any other parameters that should read by the input file will cost 10 points.
8. 5% of the grade will be based on good coding style and meaningful comments.
9. Your program should generate the output to an output file mentioned in the command line arguments.
10. Please ask questions regarding assignment and submission instruction to TA, don't assume anything.
11. Your code will be compiled and executed, if your code does not compile or produce any runtime errors such as segmentation faults or bus errors, you will get zero.

Late submission Policy

All late assignments will be penalized 5% per day (including weekends and holidays) no assignment will be accepted, if it is more than 3 days late.

Plagiarism Policy

All your code will be checked for similarity to other submissions using Moss. Programmers have an uncanny knack of reproducing the same code that they have seen before. So you are advised not to look at each other's code.

Please recall that, upon detection of plagiarism, 0 will be given to all theory and programming assignments to involved students with no exceptions.