**Assignment 3 Instructions:**

***Assignment 3 will have a different grading rubric from the standard rubric described in the syllabus.***

Theme: You will do an experiment.

Step 1:

Code a recursive algorithm for the knapsack problem in Python.

*(Note, it is preferable that you use the same data structure, templates, etc. from Assignment 1 – but if you wish to change things from the template, then that is acceptable [i.e., it will not impact your grade].)*

*{If you coded a recursive algorithm for Assignment 1, then refine and resubmit for Assignment 3.}*

Get your code to work for Data Set 10 and save an additional copy of your code for the Data Set from Step 2 (see below).

Step 2:

Create a data set with at least 30 items, and select weights, values, and capacity that will ensure a relatively difficult problem (i.e., the “enumeration” code from Assignment 1 should take at least 60 seconds to solve your created data set). Be sure to ensure that the knapsack cannot hold all of the items (i.e., it needs to be somewhat difficult; preferably too slow for enumeration and not optimal for the template file from Assignment 1).

Step 3:

Using your computer (i.e., the same computer for each run).

Run the two given template files from Assignment 1 for both Data Set 10 and your data set (from Step 2, above). Be sure to record the knapsack value and knapsack weight and the solution time for each data sets; for each method.

Run your code from your submission from Assignment 1 and from Step 1 (above) for both Data Set 10 and your data set (from Step 2, above). Be sure to record the value produced and the time. Be sure to record the knapsack value and knapsack weight and the solution time for each data sets; for each method.

**Report all of your times, knapsack values, and knapsack weights in a table**. (Note, if you really want more accurate runtimes – then take the average across 3 runs. But that is not required.)

***Just to remind you – it should be 4 different methods (both template files from Assignment 1 folder, Assignment 1 submission, and Assignment 3 recursive code).***

Step 4:

Provide the table and a write-up describing your results and make some concluding statements. It should be 1-2 pages. I’m especially interested in knowing if your submission from Assignment 1 is still “fast and near optimal” when compared to recursive methods and/or enumeration methods.

Deliverables:

Code and Data Sets for Steps 1 & 2 *(save your recursive algorithm code for both Data Set 10 and the created Data Set)*. Thus, **you should submit 2 Python files** and I (Wilck) should be able to execute them without modification since you will adjust as needed for your data set and data structure.

Write-Up (which will include the comparison table and discussion of your results/conclusions).

Grading Rubric:

Both code files run, and an appropriate recursive algorithm created to solve the Knapsack problem (50%).

Write-up is complete and table is provided; with major impacts and results identified and discussed (50%).