**Gurleen Singh Dhody**

**Assignment 2 – Algorithms (2000063241)**

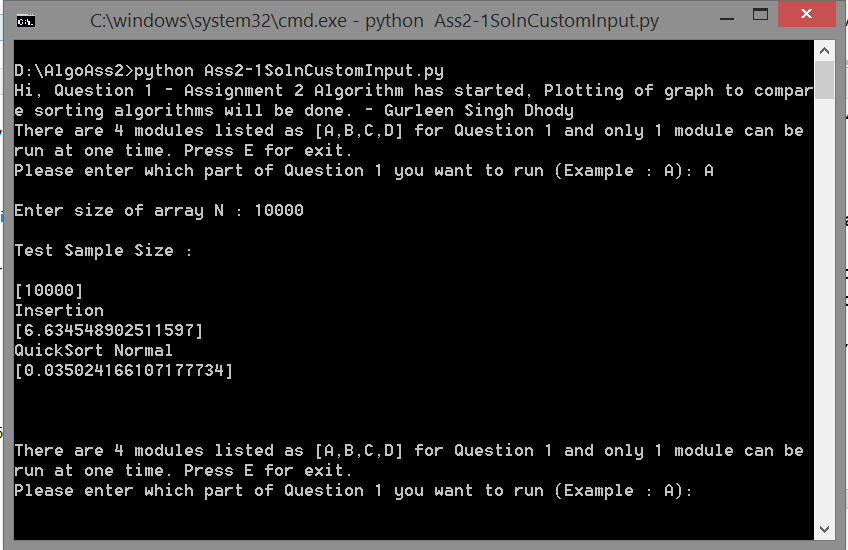
**[Note: Code was made on Python 3.3]**

Q1A) Since the input array is random the output will be different for each run. Though if the array size is very small and is nearly sorted insertion sort will overtake quicksort timing. For the random values that I generated and ran the 1st case, for the 100 input size insertion sort running time was less than quicksort. It seems Insertion sort can generally run faster if input size is 100 or less.

Q2A) Randomized quicksort is always faster than quicksort speaking theoretically. But for very large values – (input size) randomized quicksort is just a tad slow than normal quicksort. Though it is always better to use random quicksort. Since randomized quicksort has O(nlogn) expected run time, regardless of the input distribution. Whereas normal quicksort will take O(n^2) in the worst case. Even though randomized quicksort is slower, we prefer this algorithm because in the worst case scenario it will perform better than normal quicksort.

***Run the Code(Q1):***

* Ass2-1SolnGraphCode.py
  + This file generates random number and performs sorting and outputs graph
  + For building graphs matplotlib library is used <http://matplotlib.org/>
  + Python 3.3 is required to run the code
  + Please download all library files before running the code
  + On cmd you just need to type: (python Ass2-1SolnGraphCode.py)
  + The menu will ask which Test Case You want to run
  + Graphs and Excel Sheet were attached with this document
  + A screenshot of the example is shown below for   
    Ass2-1SolnCustomInput.py.



* Ass2-1SolnCustomInput.py
  + This file generates random number and performs sorting and outputs running time of algorithm comparison u choose
  + It just prints the running time of all test cases
  + Python 3.3 is required to run the code
  + On cmd type: (python Ass2-1SolnCustomInput.py)
  + Program will ask which Test Case You want to run and array (n) size

***Run the Code(Q2):***

**[Note: Please run Q2 code in Python 3.3]**

* Ass2-2Soln.py
  + Requires Python 3.3
  + On cmd you just need to type: (python Ass2-2Soln.py)
  + It will ask for the size of items bought (n)
  + It will then ask for array the quantity of each item bought
  + A screenshot for reference is given below

