Assignment 4

- Answer the questions for each of the following algorithms (as discussed in class).
 - 1. Write a naive top-down recursive algorithm to compute nth Fibonacci number.
 - 2. Write a top-down memoized version of the previous algorithm to compute nth Fibonacci number.
 - 3. Write a bottom-up dynamic programming algorithm to compute *n*th Fibonacci number. [Note: Store the results of sub-problems in an array.]
 - 4. Write a bottom-up dynamic programming algorithm to compute *n*th Fibonacci number. [Note: Don't use an array to store the results of sub-problems.]
 - 5. Write a top-down dynamic programming algorithm to compute nth Fibonacci number using linear algebra.
 - 6. Write a bottom-up dynamic programming algorithm to compute nth Fibonacci number using linear algebra.

Provide the details of Hardware/Software you used to implement the algorithms and to measure the time. Answer the following questions:

- 1. Submit the pseudocode and time/space complexity analysis of the above algorithms.
- 2. Submit the code (complete programs).
- 3. Compare the performance of the above algorithms for the following inputs. Plot a graph (X Input and Y Running Time of the Algorithms). In a single graph, you should plot all six algorithms. The values of input n are as follows:
 - -1
 - -3
 - -11
 - -107
 - -8508
 - -3033018665
 - -9743646234360185430
 - -177011382145617908261934918471444127589

- $-9404468686537716985772601572309280305961723506423561656342633485979\\029532787$
- $-\begin{array}{l} 9246489928596473588511387171174123223888880254661355091233532114148\\ 9950229823738600584381532977728037790157650018038617799318249154506\\ 1480256027030189664 \end{array}$
- $-2186655285076835579242458248058760768432746573660378248100310809093\\ 7570188630666304087119300461406495269079137225610886851037889662583\\ 8120608202261737532703779593998143967322270196415593445287444158734\\ 5998852987702534882772893162787754163095671664108778904366178041633\\ 0900685693787756842983110589060123865116694316420597930702855676385\\ 2522700574489847273651880854891897852793062520374243596076289854094\\ 5280223740467954883582400224673183343603397844562240492889243784776\\ 4205780487942416142804846159950667248017300912297619985887329672410\\ 8592475148840532690057986155941995179764937308936820602419215618313\\ 61594385484772$
- $-\ 1862242763129790521533574741956452516012712037723362157177137530069$ 02164488184337330710686018116467869488820182793491811234103434614616055121612939264158355515428724771831177485749804211163026356807654
- $-4292511349753341647424357955148085372911006942063425074989152754422\\ 7709575450764353540145614684392040985931967962232183052838088882897\\ 9954685867684580935848086283872600898555456096644214569774054882703\\ 3268225140575077496001485439027021787811764921641640883707093154877\\ 1345021677302780797059153049984248269948956437671599073523445136165$