CS/MCS 401, Summer 2016, Lowman, Midterm (HW7).
due in class Friday, Aug 5.

1. What is the optimal way to compute $A_1A_2A_3A_4A_5A_6$, where the dimensions of the matrices are: A_1:10x20, A_2:20x1, A_3:1x40, A_4:40x5, A_5:5x30, A_6:30x15

   - give the complete matrix with all $m[i, j]$ values,
   - the complete matrix s with k values, and
   - all the calculations needed to solve the problem.

2. Let $P(n)$ be the number of ways to parenthesize n matrices.

$$P(n) = \begin{cases} \sum_{k=1}^{n-1} P(k)P(n-k) & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

   (a) demonstrate how $P(n)$ is related to the Catalan numbers.

   (b) Show that $P(n) = \Omega\left(\frac{4^n}{n^{3/2}}\right)$

   (c) Show that $P(n) = \Omega\left(2^n\right)$

3. Dynamic Programming

   (a) Give a recursive algorithm that returns the nth Fibonacci number.

   (b) Give a recurrence relation for your algorithm and solve it to get a lower bound.

   (c) Give a recursion tree for the function call $f(6)$

   (d) List the two Hallmarks of Dynamic Programming and then determine if any or both are satisfied. Is this a good candidate for dynamic programming?

   (e) Give a dynamic programming algorithm using an array that stores values when they are calculated (Memoization) and later looks up the values instead of recalculating them. Give and explain a running time for your algorithm.

   (f) Repeat all of the above for an algorithm that uses the recursive definition of $n!$.

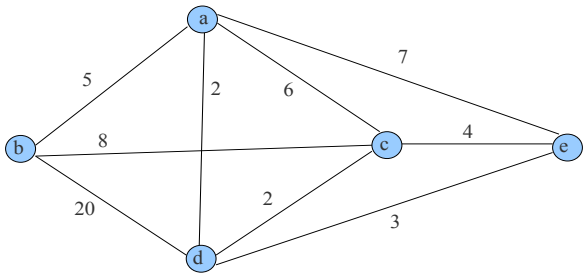
4. Write down Dijkstra's algorithm and give asymptotic running times for the algorithm. Give and explain the running times for two cases:

   - Q is implemented as an array.
   - Q is implemented as a heap.

5. Write down Prim's algorithm and give asymptotic running times for the algorithm. Give and explain the running times for two cases:

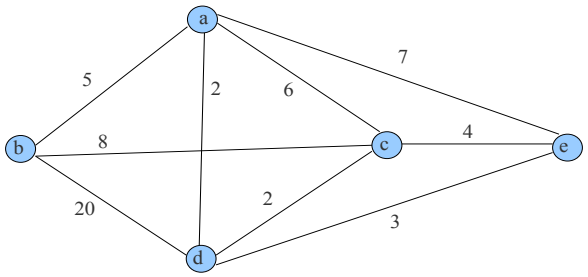
   - Q is implemented as an array.
   - Q is implemented as a heap.

6. Use the given graph for Prim's algorithm using node ***a*** as the first node.



   (a) Apply Prim's algorithm to the graph. Redraw the graph after each update. Use the the example on page 635 of CLRS as a model.

7. Use the same graph from the previous problem for Dijkstra's algorithm. Use node ***a*** as the source.



   (a) Apply Dijkstra's algorithm to the graph. Redraw the graph after each update of the distances from the source node. Use the center of the node circle to label the distances (as in textbook). Dijkstra's algorithm does not keep track of the edges for the shortest paths. Do not include them in your graphs.

   (b) After finishing with Dijkstra's algorithm, use the resulting distances to construct the actual paths. Draw one graph with all the edges in the shortest paths.

8. Chapter 34: NP-Completeness
   Give a general (not too long, not too much detail and proofs not needed) explanation of the following:

   - $\boldsymbol{P}$
   - $\boldsymbol{NP}$
   - $\boldsymbol{NP-complete}$
   - $\boldsymbol{NP-hard}$
   - Reducibility: what is a reduction
   - $\boldsymbol{SAT}$
   - $\boldsymbol{CNF}$ vs $\boldsymbol{3-CNF}$
   - $\boldsymbol{CLIQUE}$
   - Why is all of this important (or not) to a computer programmer.