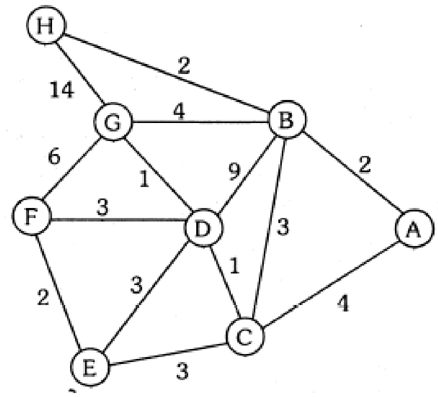
CSCI 3104: Algorithms
Homework 6

Due at **11:00am on Wednesday, October 21, 2015**. Submit your solutions electronically at moodle (name file as **LastName_FirstName_HW6.pdf**) or submit in paper before class. Make sure to include your name and student ID. Digital submission should include the Honor Code Pledge (`http://honorcode.colorado.edu/about-honor-code`), and paper submission should include your signature indicating adherence to the Honor Code Pledge.



1. Given the graph above, use Kruska's algorithm and Prim's algorithm to find the minimum spanning tree. Break ties using alphabetical order (e.g., if edges have the same cost, pick (A, D) over (A, G) and pick (A, H) over (C, F). Show the order of the edges added by each algorithm.

2. Given two sets A and B, each containing n positive integers, your goal is to reorder the value in each set such that $\prod_{i=1}^{n} a_i^{b_i}$ is maximized, where a_i and b_i are the i-th value in each set after reordering. Design a greedy algorithm and show that it is optimal.

3. A long string consists of the six characters A, B, C, D, E, F, G; they appear with frequency 21%, 11%, 8%, 17%, 5%, 23%, and 15%, respectively.
   (a) Draw the Huffman encoding tree of these six characters.
   (b) What is the Huffman encoding of these six characters?
   (c) If this encoding is applied to a string consisting of one million characters with the given frequencies, what is the length of the encoded string in bits?

4. Given n jobs, the i-th job starts at time s_i and finishes at time f_i. Two jobs are considered compatible if they do not overlap in time. Our goal is to find the maximum subset of jobs that are mutually compatible. A greedy algorithm can be designed which considers the jobs in a certain order and selects a job if and only if it is compatible with the jobs that have already been selected. For each of the following three ordering methods, determine if the greedy algorithm guarantees the optimal solution or not. If yes, show a brief proof. If not, give a counter example.
   (a) **Earliest start time**: Considers jobs in ascending order of s_i.
   (b) **Earliest finish time**: Considers jobs in ascending order of f_i.
   (c) **Shortest interval**: Considers jobs in ascending order of $f_i - s_i$.