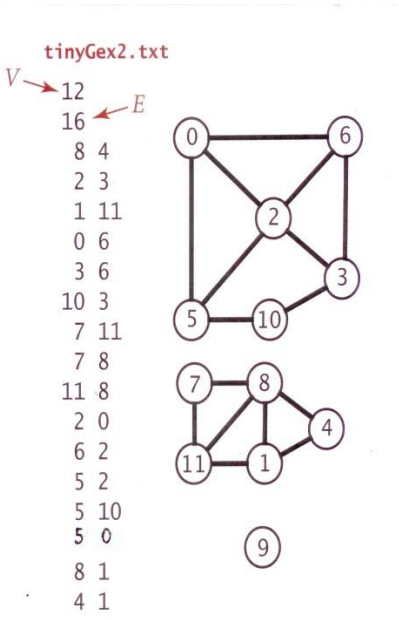
CS 2c03

Assignment #3. Due April 7 (Tuesday), 2020, 23:59 via the course' svn depository. Do not hesitate to discuss with TA or instructor all the problems as soon as you discover them.
This assignment, as previous ones, is labour consuming. Start early!

Submission instructions:

Please submit your entire assignment as a single **RAR** file. You must upload one RAR file only, including all your Java files and a single **PDF file** with the corresponding instructions on how to compile it and run some test cases. Also, put the answer of all non-programming questions in the same PDF file. Moreover, good programming style will also be marked.
Please include your MacID and student number in the PDF file.
Please use "asg#_Macid.rar" for your file name where # should be replaced by the assignment number (1,2 or 3) and Macid should be replaced by your MacID.
You must upload your solutions in the course' svn repository. Any problem with svn repository please discuss with Morteza Alipourlangouri <alipoum@mcmaster.ca>, a TA for this course.
Please check your RAR file can be extracted with no issue before submitting.

1.[5] a.[3] Draw, in the style of the figure in the text (page 524), the adjacency lists built by Graph's input stream constructor for the file tinyGex2.txt depicted below.
b.[2] Represent the graph below as adjacency matrix.



2.[6] a.[3] Show, in the style of the figure on page 533, a detailed trace of the call dfs(0) for the graph built by Graph's input stream constructor for the file tinyGex2.txt and graph above.
b.[3] Also, draw the tree represented by edgeTo[].

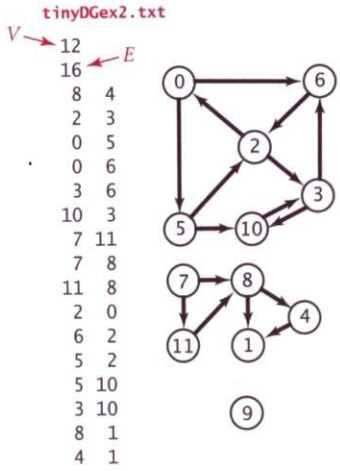
3.[6] a.[3] Show, in the style of the figure on page 533, a detailed trace of the call bfs(0) for the graph built by Graph's input stream constructor for the file tinyGex2.txt and graph above.

b.[3] Also, draw the tree represented by edgeTo[].

4.[10] Prove the following, known as the cycle property: Given any cycle in an edge weighted graph (all edge weights distinct), the edge of maximum weight in the cycle does not belong to the Minimum Spanning Tree of the graph.

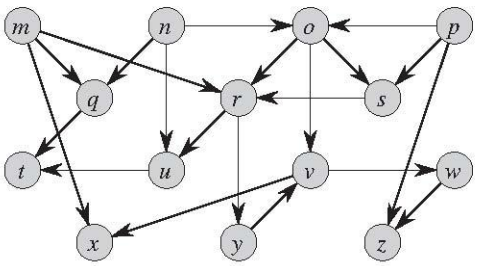
5.[5] a.[3] Draw, in the style of the figure in the text (page 524), the adjacency lists built by Digraph's input stream constructor for the file tinyDGex2.txt depicted below.

b.[2] Represent the graph below as adjacency matrix.



6.[6] Consider the connected graph above that has the set of vertices V={0,2,5,6,3,10}. Find all its strongly connected components. Use Kosaraju-Sharir algorithm. Illustrate all steps.

7.[6] Consider the DAG below:



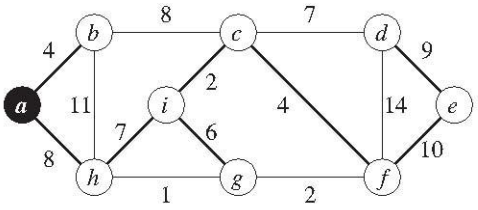
Assume that the order of vertices on the list of adjacent vertices is the alphabetical order. For example the list of adjacent vertices for the vertex m looks as follows:

$$m \rightarrow q \rightarrow r \rightarrow x$$

Provide topological order for the above DAG.

8.[8] Show that if a graph's edges all have distinct weights, the Minimum Spanning Tree is unique, but not vice versa.

9.[12] Consider the graph below.

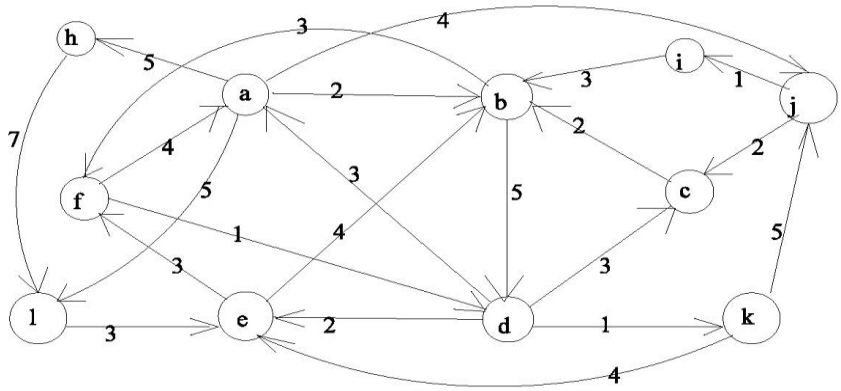


a.[4] Find Minimum Spanning Tree using Greedy Algorithm. Show all steps.

b.[4] Find Minimum Spanning Tree using Kruskal's Algorithm. Show all steps.

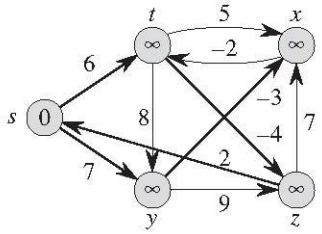
c.[4] Find Minimum Spanning Tree using Prim's Algorithm. Show all steps.

10.[8] Consider the weighted directed graph below.



Use the Dijkstra's algorithm to find the shortest paths from *a* to the other vertices. Assume that the order of vertices on the list of adjacent vertices is the alphabetical order.
Also construct the P array (recovering the paths) step by step. Recover all paths.
Assume the list of adjacent vertices implementation and with alphabetic ordering of each list.
Use heap to implement the set V– S. Show all steps.

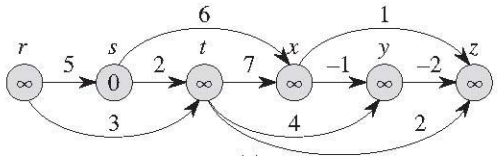
11[7] Consider the weighted graph with negative weights below.



Use the Bellman-Ford algorithm to find the shortest paths from *a* to the other vertices. Show all steps.

12.[10] The *diameter* of a digraph is the length of the maximum-length shortest path connecting two vertices. Write a DijkstraSP client that finds the diameter of a given EdgeWeightedDigraph that has nonnegative weights.

13.[7] Consider the weighted directed acyclic graph below.



Topological sort has already been applied and the result is: r, s, t, x, y, z.
Apply the second part of the algorithm for the shortest paths in DAGs, and find all shortest paths from the source r.

14.[12] a.[4] Give a trace for LSD string sort for the keys:

no is th ti fo al go pe to co to th ai of th pa

b.[4] Give a trace for MSD string sort for the keys:

no is th ti fo al go pe to co to th ai of th pa

c.[4] Give a trace for MSD string sort for the keys:

now is the time for all good people to come to the aid of

15.[8] Draw the *R*-way trie that results when the keys:

now is the time for all good people to come to the aid of

are inserted in that order into an initially empty trie (do not draw null links).

16.[8] Consider the pattern P = abbababaaabab and the string

S = abaababaabbbababaaababaabaabbabb

Compute the failure function for P, and then use Knuth-Morris-Pratt algorithm, as described in Lecture Notes 18, to verify if P appears in S. Show all steps. How many character comparisons are required by the KMP algorithm?

17.[8] Consider the pattern P = a b a c a b and the string

S = b a c a a b a c c a b a c a b a a b b

Use Boyer-Moore algorithm to verify if P appears in S. Show all steps. How many character comparisons are required by the Boyer-Moore algorithm?

18.[8] Consider the pattern P = a b a c a b and the string

S = b a c a a b a c c a b a c a b a a b b

Assume the following numerical codes for the letters a, b, and c, code(a)=1, code(b)=2, and code(c)=3, so the pattern P = a b a c a b is represented by the number 121312. Use Rabin-Karp algorithm to verify if P appears in S. Use modular hashing with R = 10 and hash(s) = s mod 997 (as in the textbook and lecture notes). How many character comparisons are required by the Rabin-Karp algorithm?

19.[5] Consider the four variable-length codes shown in the table below. Which of the codes are prefix-free? Uniquely decodable? For those that are uniquely decodable, give the encoding of 1000000000000.

symbol	code 1	code 2	code 3	code 4
A	0	0	1	1
B	100	1	01	01
C	10	00	001	001
D	11	11	0001	000

20.[8] In the style of the figure in the text, show the Huffman coding tree construction process when you use Huffman for the string "it was the age of foolishness". How many bits does the compressed bitstream require?

21.[9] What is the LZW encoding of the following inputs?

a. T O B E O R N O T T O B E

b. Y A B B A D A B B A D A B B A D O O

c. A A A A A A A A A A A A A A A A A A A A