CSC 225 Assignment 5

Trees and Graphs

University of Victoria

Due date

The submission deadline is 11:59pm on Friday July 29, 2022.

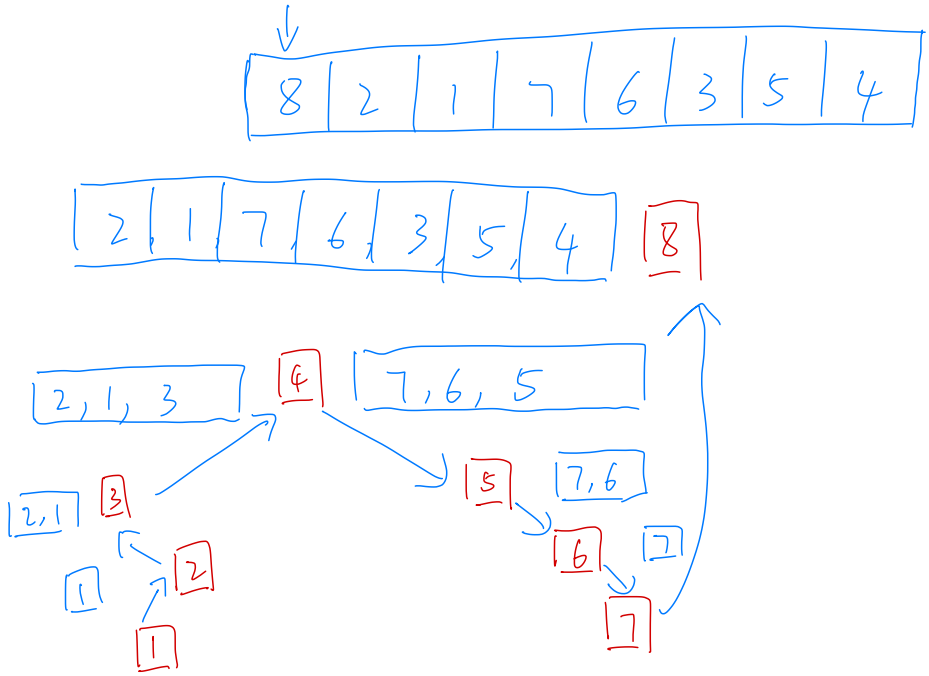
How to hand it in: Submit your **assignment5.pdf** and **GraphAlgorithms.java** files Through the Assignment 5 link on the CSC225 BrightSpace page.
IMPORTANT: The files submitted must have a **.pdf** or a **.java** extension.

Exercises

Question 1

Suppose that your Quick-Sort algorithm uses a pivot rule that always picks the FIRST element. So for any array A, regardless of size, it picks the element at A[0] as its pivot. Draw a quick-sort tree for the algorithm as it sorts the following array (without in-place partitioning):

[8 2 1 7 6 3 5 4]

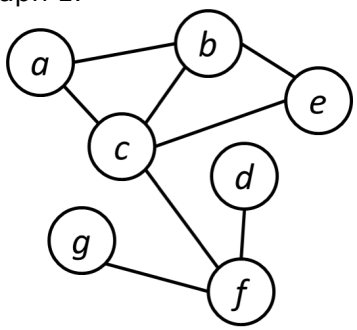


Question 2

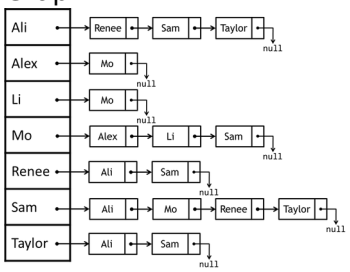
Consider the following graphs presented in three different representations:

1. a drawing of vertices and edges
2. an adjacency list
3. an adjacency matrix

Graph 1:



Graph 2:



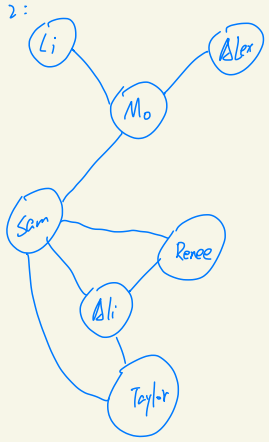
Graph 3:

	1	2	3	4	5
1	0	1	1	0	1
2	1	0	0	1	0
3	1	0	0	0	1
4	0	1	0	0	1
5	1	0	1	1	0

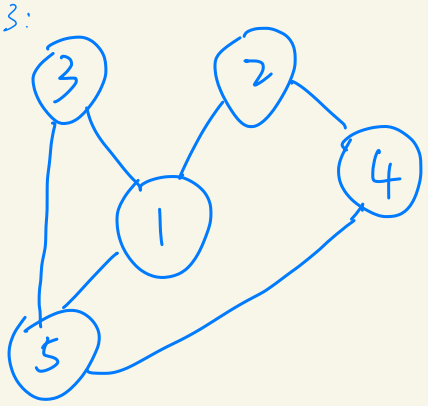
a) Draw graphs 2 and 3 as vertices and edges (similar to graph 1).
b) Convert graphs 1 and 3 into adjacency lists (similar to graph 2).
c) Convert graphs 1 and 2 into adjacency matrices (similar to graph 3).
d) What pairs of graphs are **isomorphic**? Indicate the mapping from one graph's vertex labels to the other for each pair.
e) Draw a valid spanning tree for graph 2 rooted by Sam.
f) Provide an associative array, Parents, for all nodes in the spanning tree of graph 2 rooted by Sam. Parents should consist of a set of key-value pairs where each vertex in the graph has a key, and the associated value for each key is the node's parent in the spanning tree.

a)

Graph 2:



Graph 3:



b)

Graph 1:

c -> a -> b -> e -> f -> null
a -> c -> b -> null
b -> a -> c -> e -> null
e -> c -> b -> null
f -> d -> c -> g -> null
d -> f -> null
g -> f -> null

Graph 3:

1 -> 2 -> 3 -> 5 -> null
2 -> 1 -> 4 -> null
3 -> 1 -> 5 -> null
4 -> 2 -> 5 -> null
5 -> 1 -> 3 -> 4 -> null

c)

Graph 1:

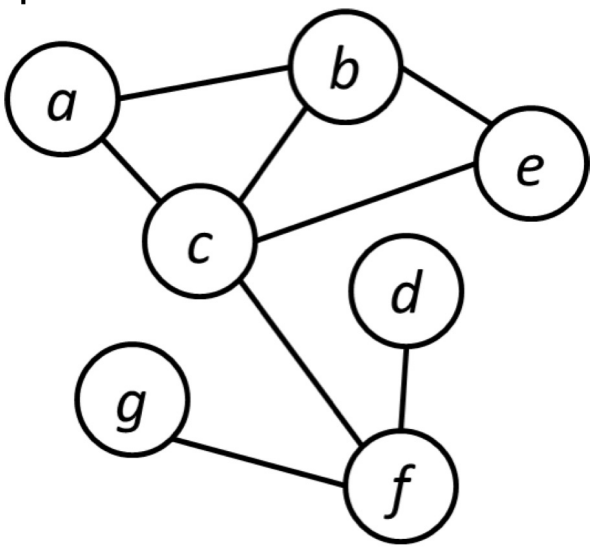
	a	b	c	d	g	e	f
a	0	1	1	0	0	0	0
b	1	0	1	0	0	1	0
c	1	1	0	0	0	1	1
d	0	0	0	0	0	0	1
e	0	1	1	0	0	0	0
g	0	0	0	0	0	0	1
f	0	0	0	1	1	0	0

Graph 2:

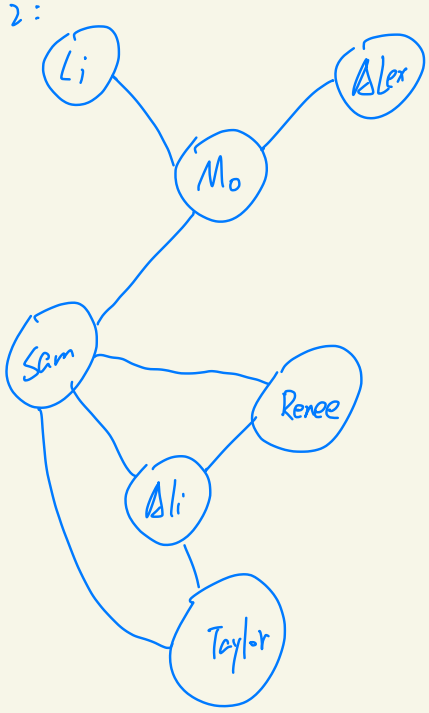
	Ali	Alex	Li	Mo	Renee	Sam	Taylor
Ali	0	0	0	0	1	1	1
Alex	0	0	0	1	0	0	0
Li	0	0	0	1	0	0	0
Mo	0	1	1	0	0	1	0
Renee	1	0	0	0	0	1	0
Sam	1	0	0	1	1	0	1
Taylor	1	0	0	0	0	1	0

d)

Graph 1:



Graph 2:



An isomorphism between Graph 1 and Graph 2

Sam = C

Renee = a

Ali = b

Taylor = c

Mo = f

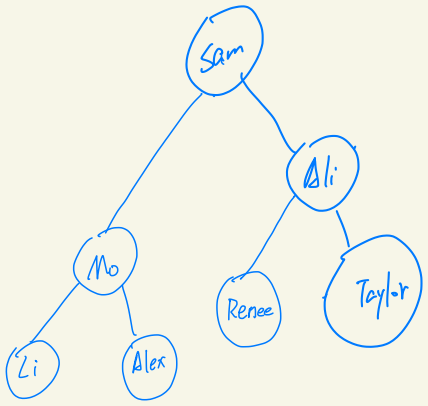
Alex = g

Li = d

∴ Graph 1 and Graph 2 are isomorphic and there is no Graph isomorphic with Graph 3 since the number of node in Graph 3 is 6 but other two Graph is 7.

e)

Graph 2:

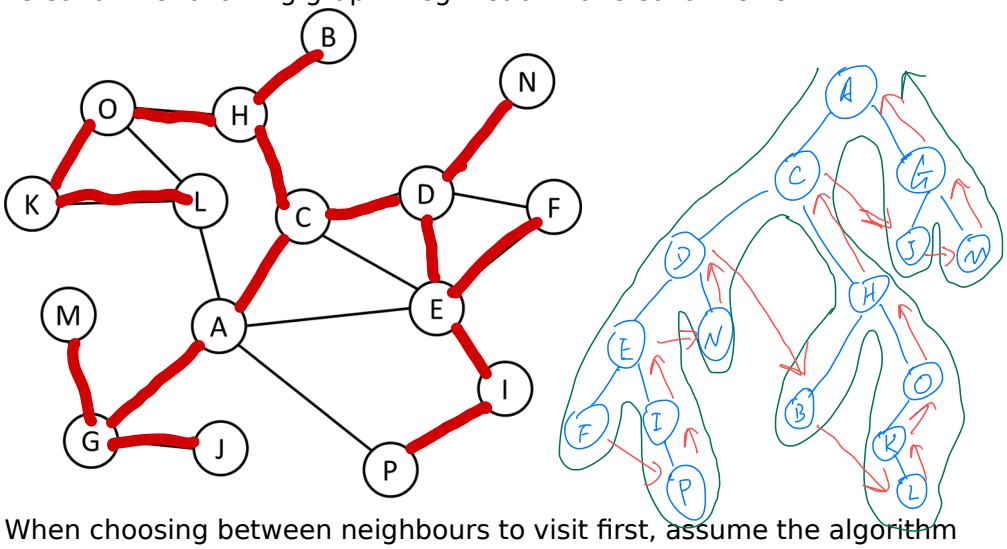


f)

key : null	Sam	Sam	Mo	Mo	Ali	Ali
Sam	Mo	Ali	Li	Alex	Renee	Taylor

Question 3

Provide the order of vertices visited for both a **pre-order** and **post-order** traversal of the following graph. Begin each traversal at vertex A.



When choosing between neighbours to visit first, assume the algorithm chooses the neighbour that comes first alphabetically.

Pre-order: A, C, D, E, F, I, P, N, H, B, O, K, L, G, J, M

Post-order: F, P, I, E, N, D, B, L, K, O, H, C, J, M, G, A

Question 4

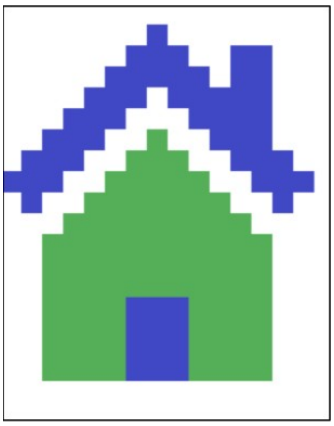
Images can be represented as 2-dimensional arrays of pixels. Usually, each pixel's Colour is based on Red, Green and Blue (RGB) components between 0 and 255. An RGB value of (255,0,0) would be red, whereas (0,255,0) is green, and (0,0,255) is blue.

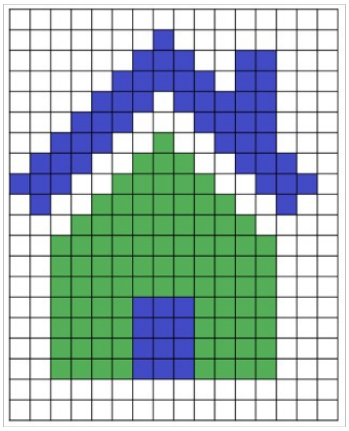
Many image processing tasks use information from neighbouring pixels. For this assignment, you will be representing images and graphs, where each vertices neighbours are adjacent pixels in the corresponding image of the same colour.

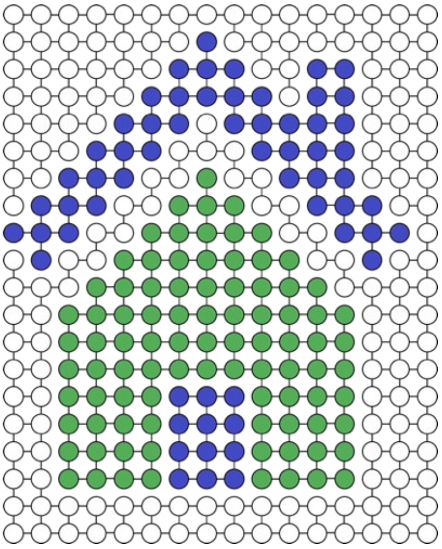
For example, look at the image below on the left depicting a house (with a blue door, blue roof and chimney, and green frame). The middle image below depicts the same image as a grid of pixels.

The image on the right shows the same image represented as a graph, with the following properties:

1. each vertex in the graph corresponds to a pixel in the image
2. each vertex is connected to adjacent pixels of the SAME colour







Your task will be to implement the flood-fill algorithm found in many image editors. To perform a flood fill, one chooses a colour and clicks a point in the image. All pixels connected to the point clicked of the same colour are then changed to the new colour.

This same process will be done using DFS and BFS traversals. When a point in the image is clicked, all connected vertices will be changed to the same colour.

The only file you need to add code to is GraphAlgorithms.java.

It will likely be necessary to read through the PixelVertex.java and PixelGraph.java files to understand how the graph has been set up, as well as the operations available to you for each vertex (PixelVertex) in the graph.

You are not expected to understand the code in the A5Tester.java file, but for those of you who are curious, feel free to experiment with it. Your solution **must** work with the current version of the A5Tester.java file. A5Tester.java is executed with the name of an image. For example, `java A5Tester house.png` opens up the testing visualization program with the house image shown previously.

There are four other sample images provided for you. All of them are very small, so if you want to animate the flood fills, you are able to without waiting too long.

Good luck!