CS211 ALGORITHMS & DATA STRUCTURES II

LAB 8

Dr. Phil Maguire

GRAPHS

PART I: Pen and paper exercise

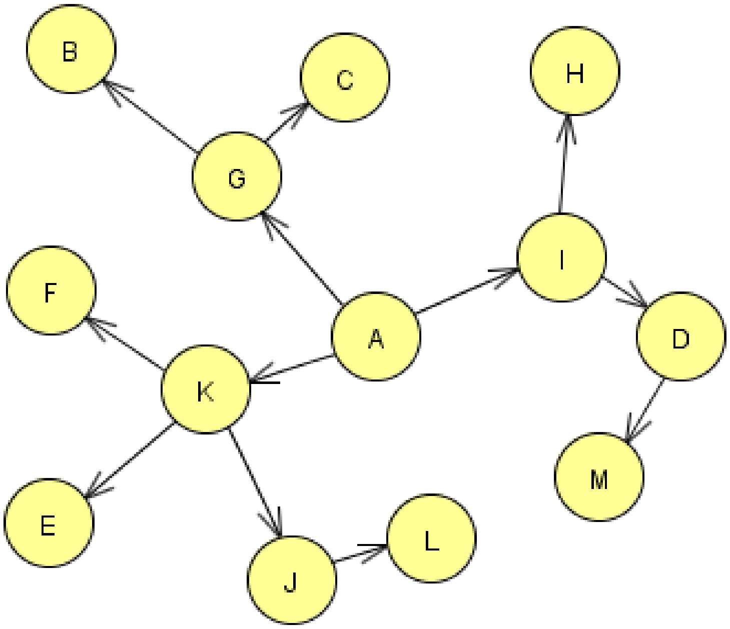
Show the orders in which breadth-first search and depth-first search would traverse the following graph starting at vertex A. Show how the contents of the search queue and search stack are updated during the searches.

Breadth-First Search

Event	Queue
Start at A	
Visit I	I
Visit K	IK
Visit G	IKG
Remove I	KG
Visit H	KGH
Visit D	KGHD
Remove K	GHD
Visit J	GHDJ
Visit E	GHDJE
Visit F	GHDJEF
Remove G	HDJEF
Visit C	HDJEFC
Visit B	HDJEFCB
Remove H	DJEFCB
Remove D	JEFCB
Visit M	JEFCBM
Remove J	EFCBM
Visit L	EFCBML
Remove E	FCBML
Remove F	CBML
Remove C	BML
Remove B	ML
Remove M	L
Remove L	Done

Depth-First Search

Event	Stack
Visit A	A
Visit I	AI
Visit H	AIH
Pop H	AI
Visit D	AID
Visit M	AIDM
Pop M	AID
Pop D	AI
Pop I	A
Visit K	AK
Visit J	AKJ
Pop J	AK
Visit E	AKE
Pop E	AK
Visit F	AKF
Pop F	AK
Pop K	A
Visit G	AG
Visit C	AGC
Pop C	AG
Visit B	AGB
Pop B	AG
Pop G	A
Pop A	Done



PART I: Programming exercise

```
public class StockData{

    public static void main(String[] args){

        FileIO io = new FileIO();
        String[] original = io.load("C:\\ stockdata.txt");
        int numrows=original.length;
        int numcols=original[0].split("\t").length;
        double[][] array = new double[numrows][numcols];

        for(int i=1;i<numrows;i++){ //load in the data
            for(int j=1;j<numcols;j++){
                array[i][j]=Double.parseDouble(original[i].split("\t")[j]);
            }
        }
        double drawdown=0;  //can we find drawdown higher than this?
        String startdate="";
        String finishdate="";
        String company="";
        for(int j=1;j<numcols;j++){ //repeat for all companies
            double current=100; //start current price at 100%
            double peak=100;    //start peak is 100%
            double trough=0;
            String localstartdate="";
            String localfinishdate="";
            String recorddate="";
            for(int i=numrows-1;i>0;i--){
```

```
//go through each day - data is backwards
                current=current+(current*(array[i][j]/100));
//change the price for today
                if(current>peak){   //if it's a record high update
                    peak=current;
                    recorddate=original[i].split("\t")[0];
//keep track of the date
                }else if(1-current/peak>trough){
//otherwise, are we lower than ever before below the current peak?
                    trough=1-current/peak;
//keep track of this super low
                    localstartdate=recorddate;
                    localfinishdate=original[i].split("\t")[0];
                }
            }
            if(trough>drawdown){
//now we've found the drawdown for this company - is it bigger than the other
companies?
                drawdown=trough;
                startdate=localstartdate;
                finishdate=localfinishdate;
                company=original[0].split("\t")[j]; //remember the company
            }
        }   //print out the overall results
        System.out.println("The company with the highest drawdown was
"+company+" which suffered a drawdown of
"+String.format("%.1f",drawdown*100)+"% between the dates of "+startdate+"
and "+finishdate);
    }
}
```