Griffin’s OH 4/9/22

COMS 1004 Introduction to Computer Science and Programming in Java

Quick Announcements

Announcements for the week of April 9th 2022

- You have made it to the last month of classes, this class ends April 28th
- Quiz 5 will be on April 14th so study now if you deem it necessary
- Problem Set 5 is due on Monday please start on it if you haven’t already done so

Topics for the Week

1. Computer Networks
2. Dijkstra’s Algorithm
3. Files and Exceptions

1

Computer Networks

Computer Networks

WAN- a collection of local-area networks (LANs) or other networks that communicate with one another. A WAN is essentially a network of networks, with the Internet the world's largest WAN.

MAN- a network with a size greater than LAN but smaller than a WAN. It normally comprises networked interconnections within a city that also offers a connection to the Internet.

LAN- a collection of devices connected together in one physical location, such as a building, office, or home. A LAN can be small or large, ranging from a home network with one user to an enterprise network with thousands of users and devices in an office or school.

Computer Networks: OSI Model

The Open Systems Interconnection (OSI) model describes seven layers that computer systems use to communicate over a network. There are seven layers to the model and attached are protocols associated with each layer:

1. Physical Layer: RS232, 100BaseTX, ISDN, 11
2. Data Link Layer: RAPA, PPP, ATM, Fiber Cables
3. Network Layer: IPv5, IPv6, ICMP, IPSEC, ARP, MPLS
4. Transport Layer: TCP, UDP
5. Session Layer: NetBIOS, SAP
6. Presentation Layer: MPEG, ASCH, SSL, TLS
7. Application Layer: HTTP, FTP, POP, SMTP, DNS

The extent of knowledge you should have over these is simply the layer names and some protocols

2

Dijkstra’s Algorithm

Dijkstra’s Algorithm

The following slides detail the actual methodology behind the algorithm, however, for the purposes of this class please format your usage of the algorithm after the wonderful video made by a fellow TA Annie: (Dijkstra’s Algorithm)

For the more curious please read on to learn more about the higher level version of Dijkstra’s Algorithm.

Dijkstra’s Algorithm

Dijkstra’s Algorithm is an algorithm that is used for finding the shortest distance, or path, from starting node to target node in a weighted graph

Here are the steps:

1. The very first step is to mark all nodes as unvisited,
2. Mark the picked starting node with a current distance of 0 and the rest nodes with infinity,
3. Now, fix the starting node as the current node,
4. For the current node, analyse all of its unvisited neighbours and measure their distances by adding the current distance of the current node to the weight of the edge that connects the neighbour node and current node,

Dijkstra’s Algorithm

5. Compare the recently measured distance with the current distance assigned to the neighbouring node and make it as the new current distance of the neighbouring node,

6. After that, consider all of the unvisited neighbours of the current node, mark the current node as visited

7. If the destination node has been marked visited then stop, an algorithm has ended, otherwise, choose the unvisited node that is marked with the least distance, fix it as the new current node, and repeat the process again from step 4.

This algorithm completes in $O(N^2)$ time

3

Files and Exceptions

Files and Exceptions

Now we want to be able to consider data from outside files. To create a file object in Java:

```
File fileName = new File(“filePath”);
```

We can then use this File Object to construct a Scanner:

```
Scanner sc = new Scanner(fileName);
```

We can then use a loop to iterate through the contents of a file:

```
while(sc.hasNextLine()){//some code to run}
```

If you want to write to a file you must construct a PrintWriter Object:

```
PrintWriter out = new PrintWriter(“outputPath”)
```

From there just .println methods to write to the file. Be sure to close the Scanner and PrintWriter when you are done processing the file prior to the program quitting

Files and Exceptions

When you detect an error condition, you should throw the appropriate exception object using the throw keyword:

```
        throw new IllegalArgumentException(“This shouldn’t happen”);
```

Sometimes you will want to handle an exception before it reaches the user for these consider a try-catch block

```
try{
    //statements that might throw an exception
}catch(//the exception you want to catch){
    //what you want to happen when exception is caught
}
```

Files and Exceptions

In Java there are two key types of exceptions: Checked and unchecked exceptions, checked exceptions can occur beyond your control while unchecked exceptions are your fault in the event that you are potentially dealing with an exception in the current method that you cannot handle add the throws clause to the method signature to tell the compiler you are aware of the exception and you expect termination to occur when the exception occurs.

```
public void readData(String filename) throws FileNotFoundException {
    //some code to run
}
```

Finally it is important to understand you can create custom exceptions the same way you make custom classes just extend RuntimeException in you Class definition when you do it!

Link to Video Drive and Other Resources

Link to Video Drive:
Video Drive

Link to My Office Hour Materials:
Useful Files