CISC-235*
Test #4
April 1, 2016

Student Number (Required) ______________________

Name (Optional) ________________________________

This is a closed book test. You may not refer to any resources.

This is a 50 minute test.

Please write your answers in ink. Pencil answers will be marked but will not be re-marked under any circumstances.

The test will be marked out of 50.

<table>
<thead>
<tr>
<th>Question 1</th>
<th>/15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 2</td>
<td>/15</td>
</tr>
<tr>
<td>Question 3</td>
<td>/15</td>
</tr>
<tr>
<td>Question 4</td>
<td>/5</td>
</tr>
</tbody>
</table>

TOTAL /50

“When you want to fool the world, tell the truth.”

Happy Birthday to Otto von Bismarck
Question 1 (15 marks)

We have studied Breadth First Search and Depth First Search. In this question we will look at a different search method called Best-First Search. Best-First Search is applicable to graphs in which each vertex has an integer value attached to it (see the example below). The search starts at some specified vertex x and searches the graph by always choosing the next reachable vertex with the highest value. We will assume the graph is connected. In pseudo-code the algorithm looks like this:

\[
\text{let } P(v) \text{ be a function that returns the value of vertex } v
\]

Best_First_Search(x):

\[
\begin{align*}
A &= \{x\} \\
\text{Chosen} &= \{\}
\end{align*}
\]

while \(|\text{Chosen}| < n\):  \hspace{1cm} # \text{n is the number of vertices}

\[
\begin{align*}
\text{let } v \text{ be the vertex in } A \text{ with the largest } P() \text{ value} \\
\text{Add vertex } v \text{ to Chosen} \\
\text{Remove vertex } v \text{ from } A \\
\text{for each neighbour } y \text{ of } v:\n\end{align*}
\]

\[
\begin{align*}
\text{if } y \text{ is not in Chosen:} \\
\text{Add } y \text{ to } A
\end{align*}
\]

If we start at vertex A, the next vertex added is E, then C, then D, then finally B

If we start at vertex D, the next vertex added is C, then B, then E, then finally A
a) What data structure would you choose to represent the graph? Why?

b) What data structure would you choose to help with the process of selecting the next vertex to add at each iteration of the while loop? Why?

c) How would you keep track of which vertices are in the set Chosen?
Question 2 (15 marks)

Let G be an undirected graph with vertex set V and edge set E.

The complement of G is defined to get the undirected graph with vertex set V and edge set E', where E' contains exactly the edges that are not in E.

a) Suppose G is represented by an adjacency matrix. How would you construct the complement of G? (this question does not require a full page to answer)
b) Suppose G is represented by a set of adjacency lists. How would you construct the complement of G?

c) Which of the two is more efficient? Even if they have the same big O complexity, which is likely to be faster?
Question 3 (15 marks)

By definition, a spanning tree of a graph contains a connecting path for each pair of vertices in the graph.

Suppose G is represented by a set of adjacency lists and a spanning tree T of G is represented by a list of edges, where each edge is in the form of a pair of vertices. Given vertices x and y, how would you find the path in T that connects x and y?

You may use any well-defined data structures you choose (which unfortunately rules out the “magically solve Question 3 in one step” structure).
Question 4 (5 marks)

Catatonia is a large country with a population of 37 million people. Suppose 95% of all residents of Catatonia submit their tax returns honestly, and that the other 5% are known to have cheated on their taxes in the past. Each year the Infernal Revenue Service wants to audit the returns of the known cheaters, but does not want to audit the returns of the honest citizens.

The IRS does not mind if a few innocent citizens are audited, but will be very upset if any cheater is not audited.

What data structure would you recommend to the government of Catatonia for the purpose of quickly deciding whether a particular tax return should be audited or not? Why?
Nearly Blank Page