Print your name and student ID, neatly in the space provided below; print your name at the upper right corner of every page. Please print legibly.

<table>
<thead>
<tr>
<th>Name:</th>
<th>Student ID:</th>
</tr>
</thead>
</table>

This is an open book exam. You are permitted to use the textbook (hard copy only), any class handouts, anything posted on the web page, any of your own assignments, and anything in your own handwriting. Foreign students may use a dictionary. Nothing else is permitted: No calculators, laptops, cell phones, Internet-enabled watches, Ipads, Ipods, communicators, GPSes, etc.!

Do all five problems in this booklet. All problems are equally weighted, so do not spend too much time on any one question.

Show your work! You will not get partial credit if the grader cannot figure out how you arrived at your answer.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
<th>Grader</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Adjacency Structures & Breadth First Search

A *fan* of order $n$ is an undirected graph on vertices $v_0, v_1, \ldots, v_n$ in which each $v_i, i > 0$, has an edge to $v_0$ and to $v_{i-1}$ and $v_{i+1}$, if they exist:

- $v_1$
- $v_2$
- $v_3$
- $\vdots$
- $v_n$

(a) Because the order of vertices in the adjacency structure lists is unimportant, many different adjacency structures are possible for a fan of order $n$. How many?

(b) We run BFS on such this graph starting at $v_1$. Label the vertices in the graph above with the the $d$-values assigned by BFS.
2. Depth First Search of Graphs

Modify the following DFS, as given in the lecture notes for November 1, to count the number of connected components in a graph. Add any statements necessary needed to count components and cross off any statements not needed for the purpose of counting components.

```java
1: function DFS(G)
2: for all u ∈ V[G] do
3: color[u] ← WHITE
4: π[u] ← NIL
5: end for
6: time ← 0
7: for all u ∈ V[G] do
8: if color[u] = WHITE then
9: DFS-visit(u)
10: end if
11: end for
12: end function

13: function DFS-visit(u)
14: color[u] ← GRAY
15: d[u] ← time ← time + 1
16: for all v ∈ Adj[u] do
17: if color[v] = WHITE then
18: π[v] ← u
19: DFS-visit(v)
20: end if
21: end for
22: color[u] ← BLACK
23: f[u] ← time ← time + 1
24: end function
```

This was suggested as a good exam problem in the lecture of November 8.
3. Graph Structure

Modify Euler’s formula for a planar graph consisting of $k$ connected components and use induction on $k$ to prove the modified version is correct. This was suggested as a good exam problem in the lecture of November 8.
4. Regular Languages

Are the following languages regular or not? Prove your answers.

(a) Strings with a multiple of three zeroes and an odd number of ones.

(b) English words used in this examination.

(c) \( L = \{a^n b^m | n \neq m \} \). (Hint: If \( L \) is regular, is \((\Sigma^* - L) \cap a^* b^*\) also regular?)
5. **Finite State Machines**

Consider the following finite state machine in which state $A$ is the starting state and state $B$ is the only accepting state:

![Finite State Machine Diagram]

(a) Describe clearly and succinctly *in words* the language recognized.

(b) Construct a regular expression for that language. Remember, a regular expression must be built from finite sets, union (+), concatenation, and Kleene closure (*).

(c) What is the answer to part (a) if we make $A$ an accepting state as well as $B$?