1. (20 Points) For each of the following recurrences, if the Master Theorem can be applied, use it to find a tight asymptotic bound on the solution. Be sure to fully justify your use of the theorem by (1) stating which case you are using, and (2) showing that the hypotheses for that case are satisfied. If the Master Theorem does not apply, prove that fact.

a. (10 Points) $T(n) = 16T(n/8) + n^{4/3}$

Solution:
Observe that $\log_8(16) = 4/3$ since $8^{4/3} = 2^4 = 16$. Thus $n^{4/3} = n^{\log_8(16)} = \Theta(n^{\log_8(16)})$, and by case 2 we have $T(n) = \Theta(n^{4/3} \log n)$.

b. (10 Points) $T(n) = 8T(n/3) + n^2$

Solution:
Note that $8 < 9 \Rightarrow \log_3(8) < 2$, so setting $\epsilon = 2 - \log_3(8)$ we have $\epsilon > 0$ and $\log_3(8) + \epsilon = 2$. Thus $n^2 = \Omega(n^2) = \Omega(n^{\log_3(8) + \epsilon})$, indicating case 3. To establish the regularity condition, pick any $c$ in the range $(8/9) \leq c < 1$. Then $8(n/3)^2 = (8/9)n^2 \leq cn^2$, for any $n$. By case 3: $T(n) = \Theta(n^2)$. ■
2. (20 Points) Let $G$ be a connected graph with $n$ vertices and $m$ edges. Use induction on $m$ to prove that $m \geq n - 1$. (Hint: you may use the following fact without proof. If an edge $e$ is removed from $G$, then the resulting graph $G - e$ is either connected, or has exactly two connected components $H_1$ and $H_2$.)

**Proof:**

I. Let $m = 0$. Then $G$, being connected, can have only one vertex. Therefore $m \geq n - 1$ reduces to $0 \geq 0$, and the base case is satisfied.

II. Let $m > 0$. Assume for any connected graph $G'$ with $|E(G')| < m$ that $|E(G')| \geq |V(G')| - 1$. We must show that $m \geq n - 1$. Pick any edge $e \in E(G)$ and remove it. By the above hint, we have two cases to consider.

**Case 1:** $G - e$ is connected.

In this case the induction hypothesis gives $m - 1 = |E(G - e)| \geq |V(G - e)| - 1 = n - 1$, whence $m \geq n > n - 1$, and therefore $m \geq n - 1$ as required.

**Case 2:** $G - e$ is disconnected.

Following the hint, $G - e$ consists of two connected components $H_1$ and $H_2$, each having fewer than $m$ edges. The induction hypothesis now guarantees $|E(H_i)| \geq |V(H_i)| - 1$ for $i = 1, 2$. Since no vertices were removed, $n = |V(H_1)| + |V(H_2)|$, and therefore

$$m = |E(H_1)| + |E(H_2)| + 1$$
$$\geq (|V(H_1)| - 1) + (|V(H_2)| - 1) + 1 \quad \text{(by the induction hypothesis)}$$
$$= (|V(H_1)| + |V(H_2)|) - 1$$
$$= n - 1$$

In this case also, $m \geq n - 1$.

The result now follows for all connected graphs by induction. □
3. (20 Points) Let $G$ be a graph with $n$ vertices, $m$ edges and $k$ connected components. Prove $m \geq n - k$. (Hint: Use the result of problem 2).

**Proof:**
Let the connected components of $G$ be $H_1, H_2, H_3, \ldots, H_k$. Suppose component $H_i$ has $n_i$ vertices and $m_i$ edges (for $i = 1, 2, \ldots, k$). Then by the result of problem 2, we have $m_i \geq n_i - 1$ (for $i = 1, 2, \ldots, k$). Summing these inequalities we get

$$m = \sum_{i=1}^{k} m_i \geq \sum_{i=1}^{k} (n_i - 1) = \sum_{i=1}^{k} n_i - \sum_{i=1}^{k} 1 = n - k,$$

i.e. $m \geq n - k$, as required.

4. (20 Points) Let $G$ be a directed graph. Prove that if $G$ contains a directed cycle, then DFS($G$) produces a back edge. Pseudo-code for DFS is included on the back page. (Hint: use the white path theorem.)

**Proof:**
Suppose $G$ contains a directed cycle, call it $C$. Let $y$ be the first vertex on $C$ to be discovered by DFS, and let $x$ be the vertex on $C$ that precedes $y$.

Since no vertex on $C$ is discovered before $y$, the vertices on $C$ from $y$ to $x$ are all white at the time $d[y]$. Thus at time $d[y]$, there is a white $y$-$x$ path in $G$. The White Path Theorem now guarantees that $x$ is a descendant of $y$ when DFS is complete. It follows that $(x, y)$ is a back edge.
5. (20 Points) Run BFS on the graph pictured on the back page of this exam, using source vertex \( s = 9 \). Pseudo-code for BFS is also included on the back page. Execute the for loop on lines 12-17 of BFS in increasing order by vertex label.

a. (14 Points) Fill in the following table, determine the order in which vertices enter the Queue, and draw the BFS Tree.

Solution:

<table>
<thead>
<tr>
<th>Adjacency List</th>
<th>Color</th>
<th>Distance</th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 4</td>
<td>w g b</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2 1 3 4 5</td>
<td>w g b</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3 2 5 6</td>
<td>w g b</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>4 1 2</td>
<td>w g b</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5 2 3 8</td>
<td>w g b</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6 3 7 8 9</td>
<td>w g b</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>7 6 9</td>
<td>w g b</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>8 5 6</td>
<td>w g b</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>9 6 7 11 12</td>
<td>g b</td>
<td>0</td>
<td>nil</td>
</tr>
<tr>
<td>10 11</td>
<td>w g b</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>11 9 10 12</td>
<td>w g b</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>12 11 9</td>
<td>w g b</td>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>

Queue | 9 6 7 11 12 3 8 10 2 5 1 4

BFS Tree:

b. (6 Points) Determine the shortest 9-4 path found by BFS, and the shortest 9-10 path found by BFS. Find a shortest 9-5 path not found by BFS.

Solution:

Shortest 9-4 path found by BFS: 9 6 3 2 4
Shortest 9-10 path found by BFS: 9 11 10
Shortest 9-5 path not found by BFS: 9 6 8 5
Problem 5 refers to the following graph (you may tear off this page and keep it)

The following pseudo-code is included for reference.

**BFS(G, s)**
1. for all \( x \in V(G) - \{s\} \)
2. \( \text{color}[x] = \text{white} \)
3. \( d[x] = \infty \)
4. \( p[x] = \text{NIL} \)
5. \( \text{color}[s] = \text{gray} \)
6. \( d[s] = 0 \)
7. \( p[s] = \text{NIL} \)
8. \( Q = \emptyset \)
9. Enqueue\((Q, s)\)
10. while \( Q \neq \emptyset \)
11. \( x = \text{Dequeue}(Q) \)
12. for all \( y \in \text{adj}[x] \)
13. if \( \text{color}[y] == \text{white} \)
14. \( \text{color}[y] = \text{gray} \)
15. \( d[y] = d[x] + 1 \)
16. \( p[y] = x \)
17. Enqueue\((Q, y)\)
18. \( \text{color}[x] = \text{black} \)

**DFS(G)**
1. for all \( x \in V(G) \)
2. \( \text{color}[x] = \text{white} \)
3. \( p[x] = \text{nil} \)
4. \( \text{time} = 0 \)
5. for all \( x \in V(G) \)
6. if \( \text{color}[x] == \text{white} \)
7. \( \text{Visit}(x) \)

**Visit(x)**
1. \( d[x] = (+ + \text{time}) \)
2. \( \text{color}[x] = \text{gray} \)
3. for all \( y \in \text{adj}[x] \)
4. if \( \text{color}[y] == \text{white} \)
5. \( p[y] = x \)
6. \( \text{Visit}(y) \)
7. \( \text{color}[x] = \text{black} \)
8. \( f[x] = (+ + \text{time}) \)