Problem 1 (10 points). For each pair of functions $f$ and $g$ in the following table, indicate whether $f = O(g)$, $f = \Omega(g)$ and $f = \Theta(g)$ respectively. Justify your answer for the question “whether $\lceil \sqrt{10n + 100} \rceil = O(n)$?”, using the definition of the $O$-notation.

<table>
<thead>
<tr>
<th>$f(n)$</th>
<th>$g(n)$</th>
<th>$O$</th>
<th>$\Omega$</th>
<th>$\Theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_{10} n$</td>
<td>$\log_2 (n^3)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lceil \sqrt{10n + 100} \rceil$</td>
<td>$n$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n^3 - 100n$</td>
<td>$10n^2 \log n$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Problem 2 (16 points).

(2a) (4 points). Given an array $A$ of $n$ integers, we need to check if there are two integers in the array with summation equaling 0. Consider the following simple algorithm:

1: for $i \leftarrow 1$ to $n - 1$ do
2: for $j \leftarrow i + 1$ to $n$ do
3: if $A[i] + A[j] = 0$ then return yes
4: return no.

Give a tight upper bound on the running time of the algorithm.

(2b) (12 points). Now suppose we have the same problem as (2a) except that the array $A$ is sorted in non-decreasing order. Consider the following algorithm:

1: $i \leftarrow 1, j \leftarrow n$
2: while $i < j$ do
3: if $A[i] + A[j] = 0$ then return yes
4: if $A[i] + A[j] < 0$ then $i \leftarrow i + 1$ else $j \leftarrow j - 1$
5: return no
Briefly argue about the correctness of the algorithm and give a tight upper bound on the running time of the algorithm.

Problem 3 (24 points).

(3a) (12 points). A cycle in an undirected graph $G = (V, E)$ is a sequence of $t \geq 3$ different vertices $v_1, v_2, \cdots, v_t$ such that $(v_i, v_{i+1}) \in E$ for every $i = 1, 2, \cdots, t - 1$ and $(v_t, v_1) \in E$. Given the linked-list representation of an undirected graph $G = (V, E)$, design an $O(n + m)$-time algorithm to decide if $G$ contains a cycle or not.

(3b) (12 points). A cycle in a directed graph $G = (V, E)$ is a sequence of $t \geq 2$ different vertices $v_1, v_2, \cdots, v_t$ such that $(v_i, v_{i+1}) \in E$ for every $i = 1, 2, \cdots, t - 1$ and $(v_t, v_1) \in E$. Given the linked-list representation of a directed graph $G = (V, E)$, design an $O(n + m)$-time algorithm to decide if $G$ contains a cycle or not.

Figure 1: Cycles in undirected and directed graphs. (1, 2, 5, 3) is a cycle in the undirected graph. (1, 2, 5, 6, 7, 3) is a cycle in the directed graph. However, (1, 2, 5, 8, 3) is not a cycle in the directed graph.

Remark On a cycle of a directed graph, the directions of the edges have to be consistent. See Figure 1. So, converting a directed graph to an undirected graph and then using algorithm for (3a) does not give you a correct algorithm for (3b).