CSCI 3333
Data Structures
Summer 2008
Suggested Solution to Mid-Term Examination

(1)

(a) true
(b) true
(c) false
(d) false

(2) For example:

Algorithm Boolean equalArray(a, b)
Input a, b: int arrays
Output: whether the arrays are equal.

```
if length(a) != length(b)
        return false;
else
        return equalArrayWithUpperRange(a, b, length(a))
end if;
```

Algorithm Boolean equalArrayWithUpperRange(a, b, upperIndex)
Input a, b: int arrays; upperIndex: upper range of the arrays to be checked.
Output: whether the arrays a[0,upperIndex-1] and b[0,upperIndex-1] are equal.

```
if upperIndex <= 0 return false;
if a[upperIndex-1] != b[upperIndex-1]
        return false;
else
        return equalArrayWithUpperRange(a, b,upperIndex-1);
end if;
```

(3) Since $f(n) = O(n^2)$ and $g(n) = O(n^3)$),
there exists c_f and n_{fo} such that
$f(n) <= c_f n^2$ for $n >= n_{fo}$
and
c_g and n_{go} such that
$g(n) <= c_g n^3$ for $n >= n_{go}$

Select $c = c_f * c_g$ and $n_o = \max(n_{fo}, n_{go})$.
Thus, for $n > n_o$,

$f(n) * g(n) <= c_f n^2 * c_g n^3 = c * n^5$

(4) For example:

```
FloatList::ListNode* FloatList::findTail() {
	ListNode *curr = head;

	if (head == NULL) return NULL;
	while (curr->next != NULL) curr = curr-> next;
	return curr;
}

void FloatList::appendTailNode(float num)
{	ListNode *newNode;
	// Allocate a new node & store num
	newNode = new ListNode;
	newNode->value = num;
	newNode->next = NULL;

	if (head==NULL)
	{	// empty list
		head = newNode;
	}
	else
	{	ListNode *tail = findTail();
		tail->next = newNode;
	}
}

int FloatList::size() {
	int result = 0;
	ListNode *curr = head;
	while (curr != NULL) {
		curr = curr-> next;
		result++;
	}
	return result;
}
```

(5) For example, in Java, with a time complexity of O(n):

```
private static int doubleFactorial(int n) {
	if (n < 0) return -1;
	if (0 <= n && n < 2) return 1;
	return n * doubleFactorial(n-2);
}
```