

Chapter 9: Pages 287-288

1. Exercise #7 (20 pts)

For each of the classes `ArrayList` and `LinkedList`, implement the method `contains`, as described in Exercise 7 of Chapter 8.

`ArrayList:`

```
template<class ItemType>
template<class ItemType>
bool ArrayList<ItemType>::contains(ItemType itemToLookFor) const {
    bool checkIfContains = false;
    for (int i = 0; !checkIfContains && (i < itemCount); i++) {
        if (itemToLookFor==items[i])
            checkIfContains = true;
    }
    return checkIfContains;
} //end contains
```

`LinkedList:`

```
template<class ItemType>
bool LinkedList<ItemType>::contains(ItemType itemToLookFor) const {
    bool checkIfContains = false;
    Node<ItemType>* curPtr = headPtr;
    while (curPtr != nullptr && !checkIfContains) {
        if (curPtr->getItem() != itemToLookFor)
            curPtr = curPtr->getNext();
        else
            checkIfContains = true;
    }
    return checkIfContains;
}
```

1. Exercise #1 a, b, c, d, e, f (6 pts)

Using Big O notation, indicate the time requirement of each of the following tasks in the worst case. Describe any assumptions that you make.

- a. $O(n)$
- b. $O(n^2)$
- c. $O(n)$
- d. $O(n)$
- e. $O(1)$
- f. $O(n)$
- g. $O(n)$

2. **Exercise #3** a, e, f, i (4 pts)

Using Big O notation, indicate the time requirement of each of the following tasks in the worst case.

a. $O(n)$

e. $O(1)$

f. $O(n)$

i. $O(n)$

3. **Exercise #5** (5 points)

Consider the following C++ function f , which calls the function `swap`. Assume that `swap` exists and simply swaps the contents of its two arguments. Do not be concerned with f 's purpose. How many comparisons does f perform?

for loop n and compares i with j so an $n + 1$ comparisons

$$T(n) = n \cdot (n+1) = n^2 + n$$

$$O(n) = n^2$$