



## [CS302-Data Structures] Quiz 4

*Instructor:* Kostas Alexis

*Teaching Assistants:* Shehryar Khattak, Mustafa Solmaz

Fall 2018 Semester

**Student First Name** \_\_\_\_\_ **Student Last Name** \_\_\_\_\_  
**Student NSHE ID** \_\_\_\_\_ **Student E-mail** \_\_\_\_\_

**Q1 (ADT List):** Give an advantage and a disadvantage of using containment in the implementation of the class SortedListHasA

**A1:**

*Advantage:* It is easier to write because the underlying list does most of the work and has been debugged.

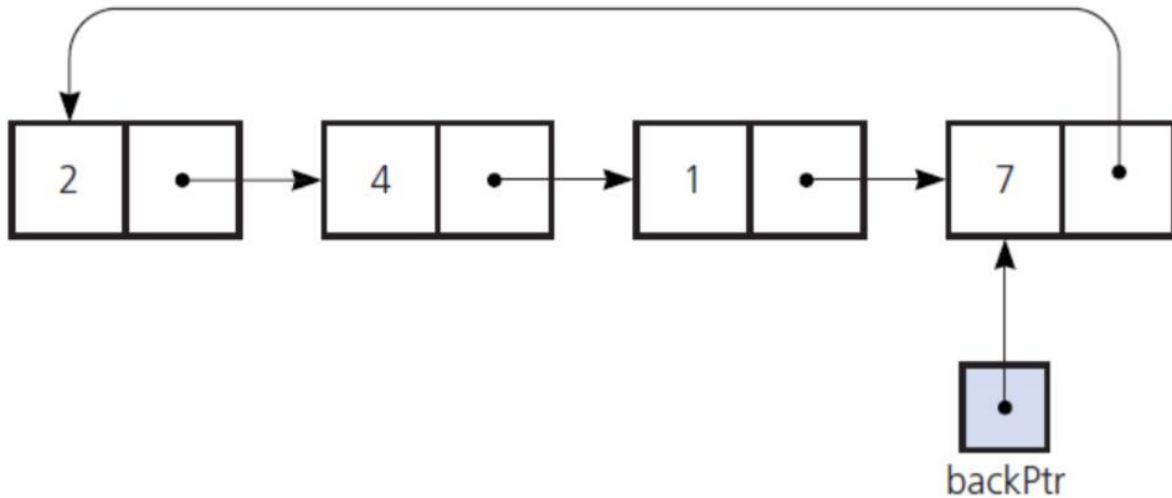
*Disadvantage:* It is less efficient than an implementation that does not call and is not restricted to the methods of the ADT list.

**Q2 (ADT Queue):** Why is a tail pointer desirable when you use a linear chain of linked nodes to implement a queue?

**A2:**

Because additions to a queue occur at its end, you would want to add nodes to the end of the linked chain. The tail pointer enables you to make such additions efficiently. With only a head pointer, you would have to traverse the entire chain each time you wanted to add to the end of the chain.

**Q3 (ADT Queue):** If you use a circular chain that has only a tail pointer as shown in the Figure below, how do you access the data in the first node.



**A3:**

First, you get a pointer to the first node by executing either of the following statements:

```
std::shared_ptr<Node<ItemType>> frontPtr = backPtr->getNext();
```

or

```
auto frontPtr = backPtr->getNext();
```

Then the data in the first node is

```
frontPtr->getItem()
```