Linked Lists

CS 240

Alex Vondrak

ajvondrak@csupomona.edu

Winter 2012

Multiple Choice Question

What is the O-complexity of a single array access (i.e., accessing a random array index)?

(A) $O(n^2)$

(B) $O(n)$

(C) $O(1)$

(D) None of the above

Multiple Choice Question

What is the O-complexity of iterating through the elements of an array, performing some constant-time operations on each item?

(A) $O(n^2)$

(B) $O(n)$

(C) $O(1)$

(D) None of the above

Multiple Choice Question

Which of the following describes how array elements are arranged in a computer's memory?

(A) One after the other

(B) Scattered randomly

(C) Some combination of the above

(D) None of the above

Multiple Choice Question

We said array iteration was $O(n)$, though random access is $O(1)$.
Does arranging array elements one after the other in memory affect the O-complexity for iteration?
That is, if we didn't arrange arrays sequentially, could iteration still be $O(n)$?

(A) Yes—as long as we knew where to look

(B) No—accessing some remote slot of memory takes too much time

(C) I don't know

Multiple Choice Question

Suppose we're iterating through a sequence of elements.
At any one particular spot, how do we know where in memory to go for the next element in the sequence?

(A) If the elements are arranged sequentially, the next element is at the next memory address

(B) If the elements are not arranged sequentially, we don't necessarily know

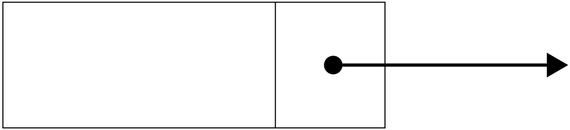
(C) Both of the above

(D) None of the above

Linked List

Definition

A linked list represents a sequence of elements by a group of nodes



A node (or cons cell) is an object that consists of two parts, which go by several different names:

- `data`/`link`
- `head`/`tail`
- `first`/`rest` (`fst`/`rst`)
- `car`/`cdr`

Multiple Choice Question

Nodes can be represented by Java objects:

```
class Node<E> {
    ⋮
}
```

What fields should the class have?

(A) `E data` and `E link`

(B) `E[] data` and `E link`

(C) `E data` and `int link`

(D) `E data` and `Node<E> link`

Multiple Choice Question

Nodes can be represented by Java objects:

```
class Node<E> {
    ??? E data;
    ??? Node<E> link;

    ⋮
}
```

Should the fields be **public** or **private**?

(A) **public**

(B) **private**

(C) One should be **public**, the other **private**

(D) They shouldn't be **public** or **private**

Multiple Choice Question

Nodes can be represented by Java objects:

```
class Node<E> {
    private E data;
    private Node<E> link;

    ⋮
}
```

What other methods do we need?

(A) A constructor

(B) Getters

(C) Setters

(D) All of the above

The Node<E> class

```
class Node<E> {
    public E data;
    public Node<E> link;

    public Node(E data, Node<E> link) {
        this.data = data;
        this.link = link;
    }
}
```

Multiple Choice Question

How do you suppose we make a linked list of integers with one element (say 10)?

(A) `new Node<Integer>(10)`

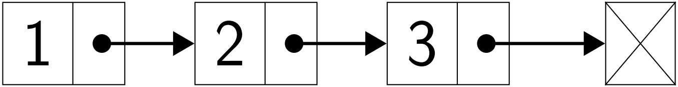
(B) `new Node<Integer>(10, null)`

(C) `new Node<Integer>(10, new Node<Integer>(null))`

(D) `new Node<Integer>(10, new Node<Integer>(null, null))`

Multiple Choice Question

How do you suppose we make a linked list of integers with three elements (say 1, 2, and 3)?



(A) `new Node<Integer>(1, 2, 3)`

(B) `new Node<Integer>(1, new Node<Integer>(2, 3))`

(C)
```
new Node<Integer>(1,
    new Node<Integer>(2,
        new Node<Ineger>(3, null)))
```

(D) None of the above

Multiple Choice Question

Suppose we have the following.

```
Node<Integer> head = new Node<Integer>(2,
                         new Node<Integer>(3, null));
```

Which of the following is the best way to add the element 1 to the front of this list?

(A) `head = new Node<Integer>(1, head);`

(B) `head = new Node<Integer>(1, head.link);`

(C) `head.data = 1; head.link = head.link.link;`

(D)
```
head = new Node<Integer>(1,
           new Node<Integer>(2,
              new Node<Ineger>(3, null)));
```

Multiple Choice Question

Suppose we have the following.

```
Node<Integer> head = new Node<Integer>(2,
                       new Node<Integer>(3, null));
```

Which of the following is the best way to remove the front element of this list?

(A) `head.data = null;`

(B) `head.link = null;`

(C) `head.link = head.link.link;`

(D) `head = head.link;`

Multiple Choice Question

Suppose we have access to the `Node<Integer> head` of an arbitrary linked list, and we want to iterate through the list to count how many 5s are in it.

```
int fives = 0;
for (?; ?; ?) {
    if (? == 5) {
        fives++;
    }
}
```

What should the iteration variable be?

(A) `int i = 0`

(B) `Node<Integer> cursor = head`

(C) `Node<Integer> cursor = 0`

(D) None of the above

Multiple Choice Question

Suppose we have access to the Node<Integer> head of an arbitrary linked list, and we want to iterate through the list to count how many 5s are in it.

```
int fives = 0;
Node<Integer> cursor;
for (cursor = head; ?; ?) {
    if (? == 5) {
        fives++;
    }
}
```

What should the “while” condition be?

(A) cursor != **null**

(B) cursor.link != **null**

(C) cursor.data != **null**

(D) None of the above

Multiple Choice Question

Suppose we have access to the Node<Integer> head of an arbitrary linked list, and we want to iterate through the list to count how many 5s are in it.

```
int fives = 0;
Node<Integer> cursor;
for (cursor = head; cursor != null; ?) {
   if (? == 5) {
      fives++;
   }
}
```

What should the “increment” action be?

(A) cursor = **this**.nextNode(cursor) (need new method)

(B) cursor.link = cursor.link.link

(C) cursor = cursor.link

(D) None of the above

Multiple Choice Question

Suppose we have access to the `Node<Integer> head` of an arbitrary linked list, and we want to iterate through the list to count how many 5s are in it.

```
int fives = 0;
Node<Integer> cursor;
for (cursor = head; cursor != null; cursor = cursor.link) {
   if (? == 5) {
      fives++;
   }
}
```

How can we access the current position's data?

(A) `cursor`

(B) `cursor.data`

(C) `head.data`

(D) `cursor.link.data`

Multiple Choice Question

Suppose we have access to the `Node<Integer> head` of an arbitrary linked list, and we want to iterate through the list to count how many 5s are in it.

```
int fives = 0;
Node<Integer> cursor;
for (cursor = head; cursor != null; cursor = cursor.link) {
   if (cursor.data == 5) {
      fives++;
   }
}
```

What is the complexity of this bit of code?

(A) $O(1)$

(B) $O(n)$

(C) $O(n^2)$

(D) None of the above