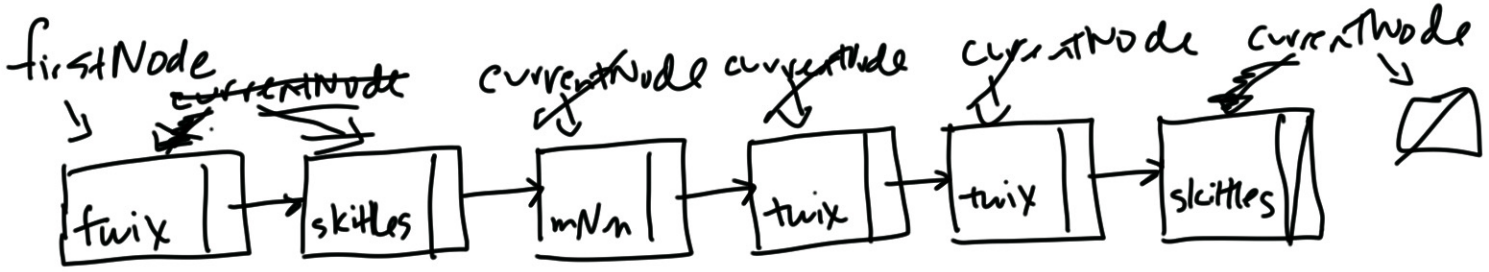
```java
    /**
     * Counts the number of times a given entry appears in this bag.
     * Can loop until null and don't need to count
     *
     * @param anEntry
     *            The entry to be counted.
     * @return The number of times anEntry appears in this bag.
     */
    public int getFrequencyOf(T anEntry) {
        int frequency = 0;
        Node<T> currentNode = firstNode;
        while ((currentNode != null)) {
            if (anEntry.equals(currentNode.getData())) {
                frequency++;
            } // end if

            currentNode = currentNode.getNext();
        } // end while

        return frequency;
    } // end getFrequencyOf
```



numberOfEntries

6

anEntry

twix

frequency

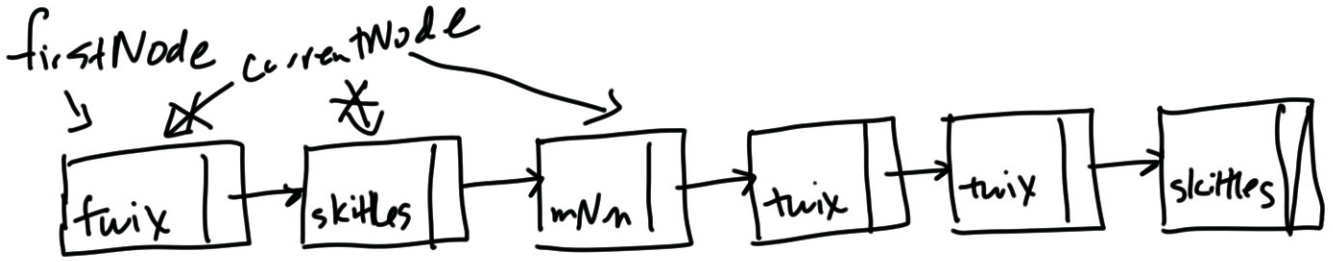
0

1

2

3

```
105    /**
106     * Tests whether this bag contains a given entry.
107     *
108     * @param anEntry
109     *            The entry to locate.
110     * @return True if the bag contains anEntry, or false otherwise.
111     */
112    public boolean contains(T anEntry) {
113        boolean found = false;
114        Node<T> currentNode = firstNode;
115
116        while (!found && (currentNode != null)) {
117            if (anEntry.equals(currentNode.getData()))
118                found = true;
119            else
120                currentNode = currentNode.getNext();
121        } // end while
```



numberOfEntries
6

anEntry
mNm

found
~~false~~
true