private static <T> void moveToFront(Queue<T> q, T x) {

assert q != null : "Violation of: q is not null";

Queue<T> leftPart = q.newInstance();

Queue<T> rightPart = q.newInstance();

while (q.length() != 0) {

T element = q.dequeue();

if (element.equals(x)) {

leftPart.enqueue(element);

} else {

rightPart.enqueue(element);

}

}

leftPart.append(rightPart);

q.transferFrom(leftPart);

}

/\*\*

\* Test addition to a non-empty set.

\*/

@Test

public final void testAddElement() {

Set<String> testSet = this.createFromArgsTest("apple", "banana");

Set<String> expectedSet = this.createFromArgsRef("cherry", "apple", "banana");

testSet.add("cherry");

assertEquals(testSet, expectedSet);

}

/\*\*

\* Test removal from a non-empty set.

\*/

@Test

public final void testRemoveElement() {

Set<String> testSet = this.createFromArgsTest("cherry", "apple", "banana");

Set<String> expectedSet = this.createFromArgsRef("apple", "banana");

testSet.remove("cherry");

assertEquals(testSet, expectedSet);

}

/\*\*

\* Test removing any element from a non-empty set.

\*/

@Test

public final void testRemoveAnyElement() {

Set<String> testSet = this.createFromArgsTest("cherry", "apple", "banana");

Set<String> expectedSet = this.createFromArgsRef("cherry", "apple", "banana");

String removedElement = testSet.removeAny();

assertTrue(expectedSet.contains(removedElement) && testSet.size() == expectedSet.size() - 1);

}

/\*\*

\* Test set contains an element.

\*/

@Test

public final void testContainsElement() {

Set<String> testSet = this.createFromArgsTest("cherry", "apple", "banana");

Set<String> expectedSet = this.createFromArgsRef("cherry", "apple", "banana");

String element = "apple";

assertTrue(expectedSet.contains(element) && testSet.contains(element));

}

/\*\*

\* Test the size of the set.

\*/

@Test

public final void testSetSize() {

Set<String> testSet = this.createFromArgsTest("cherry", "apple", "banana");

Set<String> expectedSet = this.createFromArgsRef("cherry", "apple", "banana");

int testSetSize = testSet.size();

int expectedSetSize = expectedSet.size();

assertTrue(testSetSize == expectedSetSize);

}