1. The picture shows two states of a ListNode p. In the lines below, write Java code to turn Before to After. **Do not** modify the data instance variable, introduce any new variables, or use the `new` keyword.

```
static boolean mystery(int[] a, int target) {
    int N = a.length;
    for (int i = 0; i < N; i += 1) {
        if (a[i] == target) {
            return true;
        }
    }
    return false;
}
```

2. Give a tight asymptotic runtime bound for `mystery` as a function of $N$, the length of the array, in the **best case, worst case, and overall**. Give a $\Theta(\cdot)$ bound if it exists. Otherwise, give both an $O(\cdot)$ and $\Omega(\cdot)$ bound.

```
static boolean mystery(int[] a, int target) {
    int N = a.length;
    for (int i = 0; i < N; i += 1) {
        if (a[i] == target) {
            return true;
        }
    }
    return false;
}
```

<table>
<thead>
<tr>
<th>Best case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst case</td>
</tr>
<tr>
<td>Overall</td>
</tr>
</tbody>
</table>