Ethan Rowell

Nicholas Hays

Line-by-line Analysis

```
63      public void changePatties(String pattyType) {
64              if (pattyType == myPattyType) return;
65              while (!myOrder.isEmpty()) {
66                      if (myOrder.peek().equals(myPattyType)) {
67                              myOrder.pop();
68                              myOrder.push(pattyType);
69                      } else {
70                              myTempStack.push(myOrder.pop());
71                      }
72              }
73              refillOrderStack();
74              myPattyType = pattyType;
75              updateRecipe();
76      }
77
78      private void refillOrderStack() {
79              while (!myTempStack.isEmpty()) {
80                      myOrder.push(myTempStack.pop());
81              }
82      }
83
84      private void updateRecipe() {
85              while (!myRecipe.myIsEmpty) {
86                      myRecipe.pop();
87              }
88              createRecipe();
89      }
```

```
37      private void createRecipe() {
38              myRecipe.push("Pickle");
39              myRecipe.push("Bun");
40              myRecipe.push("Mayonnaise");
41              myRecipe.push("Baron-Sauce");
42              myRecipe.push("Lettuce");
43              myRecipe.push("Tomato");
44              myRecipe.push("Onions");
45              if (myPattyCount > 1) {
46                      for (int i = 1; i < myPattyCount; i++) {
47                              myRecipe.push(myPattyType);
48                      }
49              }
50              myRecipe.push("Pepperjack");
51              myRecipe.push("Mozzarella");
52              myRecipe.push("Cheddar");
53              myRecipe.push(myPattyType);
54              myRecipe.push("Mushrooms");
55              myRecipe.push("Mustard");
56              myRecipe.push("Ketchup");
57              myRecipe.push("Bun");
58      }
```

Line	Code	Big-Oh
Line 64	`if (pattyType == myPattyType)`	O(1)
Line 65	`!myOrder.isEmpty()`	O(1)
Line 66	`myOrder.peek().equals(myPattyType))`	O(3)
Line 67	`myOrder.pop()`	O(3)
Line 68	`myOrder.push(pattyType)`	O(7)
Line 70	`myTempStack.push(myOrder.pop())`	O(10)
Line 73	`refillOrderStack()`	O(n)
Line 74	`myPattyType = pattyType`	O(1)
Line 88	`createRecipe()`	O(n)

Summation of Loops

The main `while` loop:

```
65                  while (!myOrder.isEmpty()) {
66                          if (myOrder.peek().equals(myPattyType)) {
67                                  myOrder.pop();
68                                  myOrder.push(pattyType);
69                          } else {
70                                  myTempStack.push(myOrder.pop());
71                          }
72                  }
```

Breakdown of constants:

`isEmpty()`	$\rightarrow C_1$
`peek()`	$\rightarrow 2 \cdot C_2$
`pop()`	$\rightarrow 3 \cdot C_3$
`push()`	$\rightarrow 7 \cdot C_4$
`push()` / `pop()`	$\rightarrow 10 \cdot C_5$

Summation:

$$\sum_{i=0}^{n-1} 23C \rightarrow 23C \cdot \sum_{i=0}^{n-1} 1 = 23Cn$$

$23Cn$ is $\boldsymbol{O(n)}$

The `refillOrderStack()` loop:

```
78          private void refillOrderStack() {
79                  while (!myTempStack.isEmpty()) {
80                          myOrder.push(myTempStack.pop());
81                  }
82          }
```

Breakdown of constants:

`isEmpty()` $\rightarrow C_1$

`push()` / `pop()` $\rightarrow 10 \cdot C_2$

Summation:

$$\sum_{i=0}^{n-1} 11C \rightarrow 11C \cdot \sum_{i=0}^{n-1} 1 = 11Cn$$

$11Cn$ is $\boldsymbol{O(n)}$

The `updateRecipe()` loop:

```
84          private void updateRecipe() {
85                  while (!myRecipe.myIsEmpty) {
86                          myRecipe.pop();
87                  }
88                  createRecipe();
```

```
37          private void createRecipe() {
38                  myRecipe.push("Pickle");
39                  myRecipe.push("Bun");
40                  myRecipe.push("Mayonnaise");
41                  myRecipe.push("Baron-Sauce");
42                  myRecipe.push("Lettuce");
43                  myRecipe.push("Tomato");
44                  myRecipe.push("Onions");
45                  if (myPattyCount > 1) {
46                          for (int i = 1; i < myPattyCount; i++) {
47                                  myRecipe.push(myPattyType);
48                          }
49                  }
50                  myRecipe.push("Pepperjack");
51                  myRecipe.push("Mozzarella");
52                  myRecipe.push("Cheddar");
53                  myRecipe.push(myPattyType);
54                  myRecipe.push("Mushrooms");
55                  myRecipe.push("Mustard");
56                  myRecipe.push("Ketchup");
57                  myRecipe.push("Bun");
58          }
```

Breakdown of constants:

`isEmpty()` $\rightarrow C_1$

`pop()` $\rightarrow 3 \cdot C_2$

Summation:

$$\sum_{i=0}^{n-1} 4C \rightarrow 4C \cdot \sum_{i=0}^{n-1} 1 = 4Cn$$

$4Cn$ is $\boldsymbol{O(n)}$ + `createRecipe()` $\boldsymbol{O(n)} \rightarrow \boldsymbol{O(n)}$.

Cost of method

Main `while` loop	$\rightarrow O(n)$
`refillOrderStack()`	$\rightarrow O(n)$
`updateRecipe()`	$\rightarrow O(n)$
Total cost:	$\boldsymbol{O(n)}$