

```
CS210 > sortcar.cpp
Project
  CS210 sources root, ~/De
    driver.cpp
    Makefile
    sortcar.cpp
    sortcar.h
    unnamed.pdf
    External Libraries
    Scratches and Consoles
Terminal: Local x + v
  Version Control Python Packages TODO Problems Terminal Services Build
  clang-tidy 27:77 LF UTF-8 4 spaces Context: <no context>

23 */
24 vector<string> SortCar::sortCarInventory(vector<string> carInventory, vector<string> promotionOrder) {
25
26
27   int j = 0; //Keeps an overall counter for the car inventory when sorting 1 operation
28
29   //Outer loop individually compares promotionOrder to each index in carInventory
30   for (unsigned long i = 0; i < (promotionOrder.size() - 1); i++){ 2 operation
31
32     //If promotionOrder matches carInventory increment to the next element in carInventory
33     if (promotionOrder.at(i) == carInventory.at(j)){ 1 operation
34       if (j < (carInventory.size() - 1)){ 1 operation
35         j++; 1 operation
36       }
37     }
38
39     //If promotionOrder does not match carInventory at a particular index
40     if (!(promotionOrder.at(i) == carInventory.at(j))){ 1 operation
41
42       //Runs from index j onwards through the rest of carInventory & compares elements to promotionOrder
43       for (int k = j; k <= (carInventory.size() - 1); k++){ N operations
44         string temp = carInventory.at(j); //Record the not matching element in carInventory 1 operation
45
46         if (promotionOrder.at(i) == carInventory.at(k)){ //If match is found swap elements in j & k 1 operation
47           carInventory.at(j) = carInventory.at(k); 1 operation
48           carInventory.at(k) = temp; 1 operation
49
50           j++; //Increment to next index in carInventory 1 operation
51         }
52       }
53     }
54   }
55   return carInventory; 1 operation
56 }
57
58
59 }
```

Key: Constant  $O(1)$

linear  $O(N)$

## Auxiliary Space

Memory input - promotionOrder : fixed sized

Car Inventory : vector of  $N$  strings

Memory (non input) -  $i, j, temp$

$$\hookrightarrow S(N) = 1 + 1 + 1 = 3$$

## Time Complexity

$$f(N) = 1 + 2 + 1 + 1 + 1 + 1 + N(1 + 1 + 1 + 1 + 1) + 1 \\ = 10 + 5N = O(N)$$