GTU Department of Computer Engineering

CSE 222 / 505 – Spring 2022

Homework 6

Ahmet USLUOGLU

# 1801042602

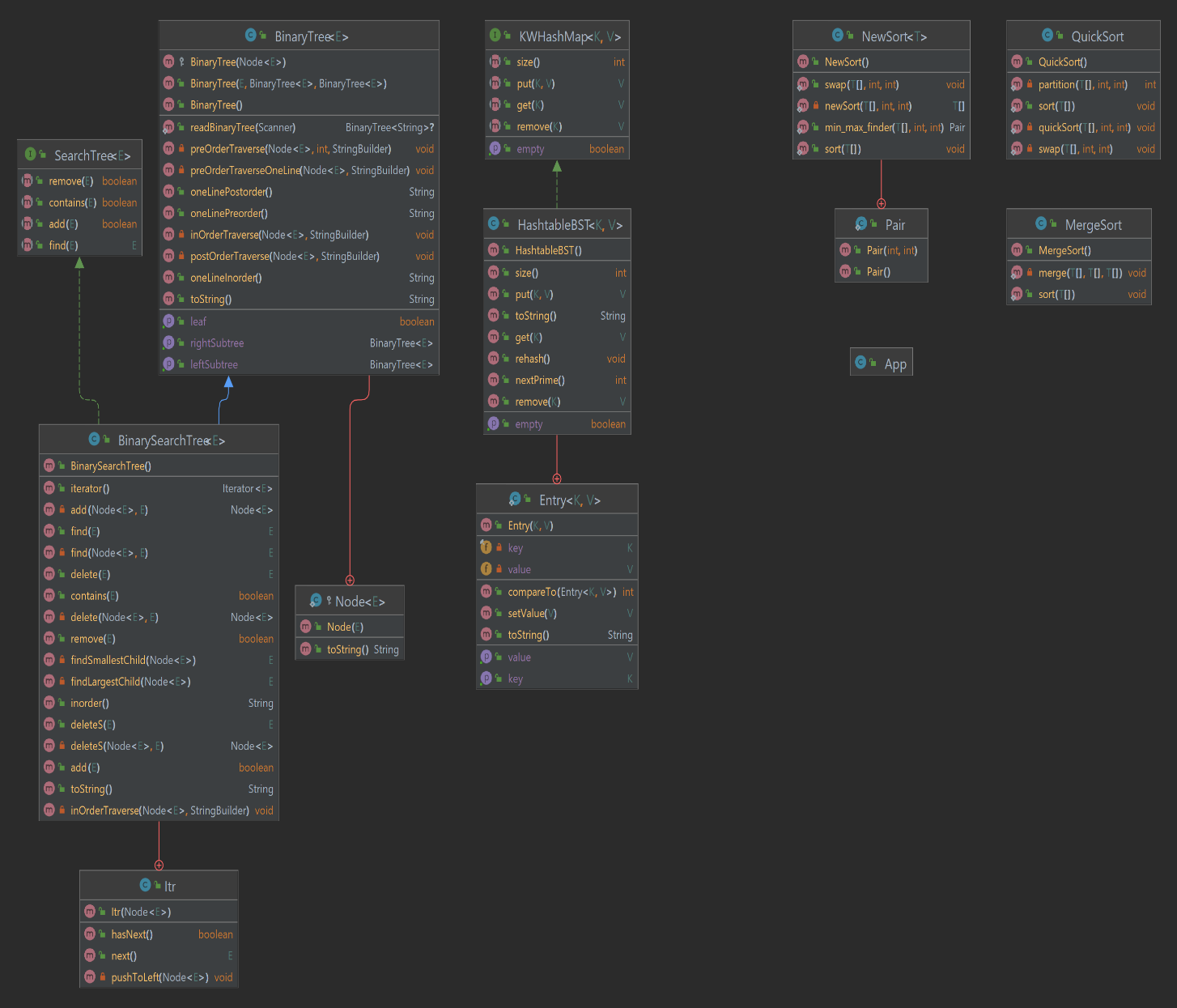
### – System Requirement

Operating System must have JDK (Java Development Kit) 11 and JRE (Java Runtime Environment) 11 or higher.

There should be enough space for storing data’s.

1. – Class Diagrams

\*Higher Resolution Version of the Class Diagram is in the files.



3. Problem - Solution Approach

Problem:

1 - ) Implement the chaining technique for hashing. U binary search trees to chain items as underlying structure mapped on the same table slot.

2-) Compare the sorting algorithms Quick Sort, Merge Sort and New Sort both empirically and theoretically.

Solution:

1-) When implementing a hash table using chaining technique if the items stack over one index more than the others, searching for an item gets longer and it approaches to linear O(n) complexity. To prevent this instead of using linked list as underlying structure , using Binary search trees makes searching time an average of O(nlogn). Thus, gives us faster average but more complex space structure.

2-) Tested 3 sort algorithms by 100 time each with 100, 1000 and 10000 elements.

1. – Test Cases

a-) Testing the hash table for part1

1 – Adding 100 elements to the hash table and measuring time.

2 – Check the isEmpty with empty hash table.

3 – Check the isEmpty with filled hash table.

4 – Remove element from hashtable.

5 – Try removing unexisting element.

6 – Add an element with same key value to update it.

b-) Testing the sort algorithms for part2

1 -) Compare 3 algorithms 100 times with 100 sized arrays.

2 -) Compare 3 algorithms 100 times with 1000 sized arrays.

3 -) Compare 3 algorithms 100 times with 10000 sized arrays.

## 5 – Running Program and Results

a) Testing the hash table for part1

1 – Adding 100 elements to the hash table and measuring time



2 – Check the isEmpty with empty hash table.



3 – Check the isEmpty with filled hash table.



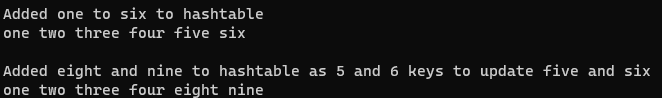
4 – Remove element from hashtable.



5 – Try removing unexisting element.

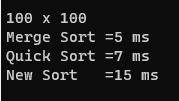


6 – Add an element with same key value to update it.

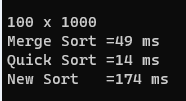


b-) Testing the sort algorithms for part2

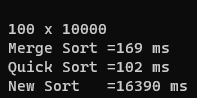
1 -) Compare 3 algorithms 100 times with 100 sized arrays.



2 -) Compare 3 algorithms 100 times with 1000 sized arrays.

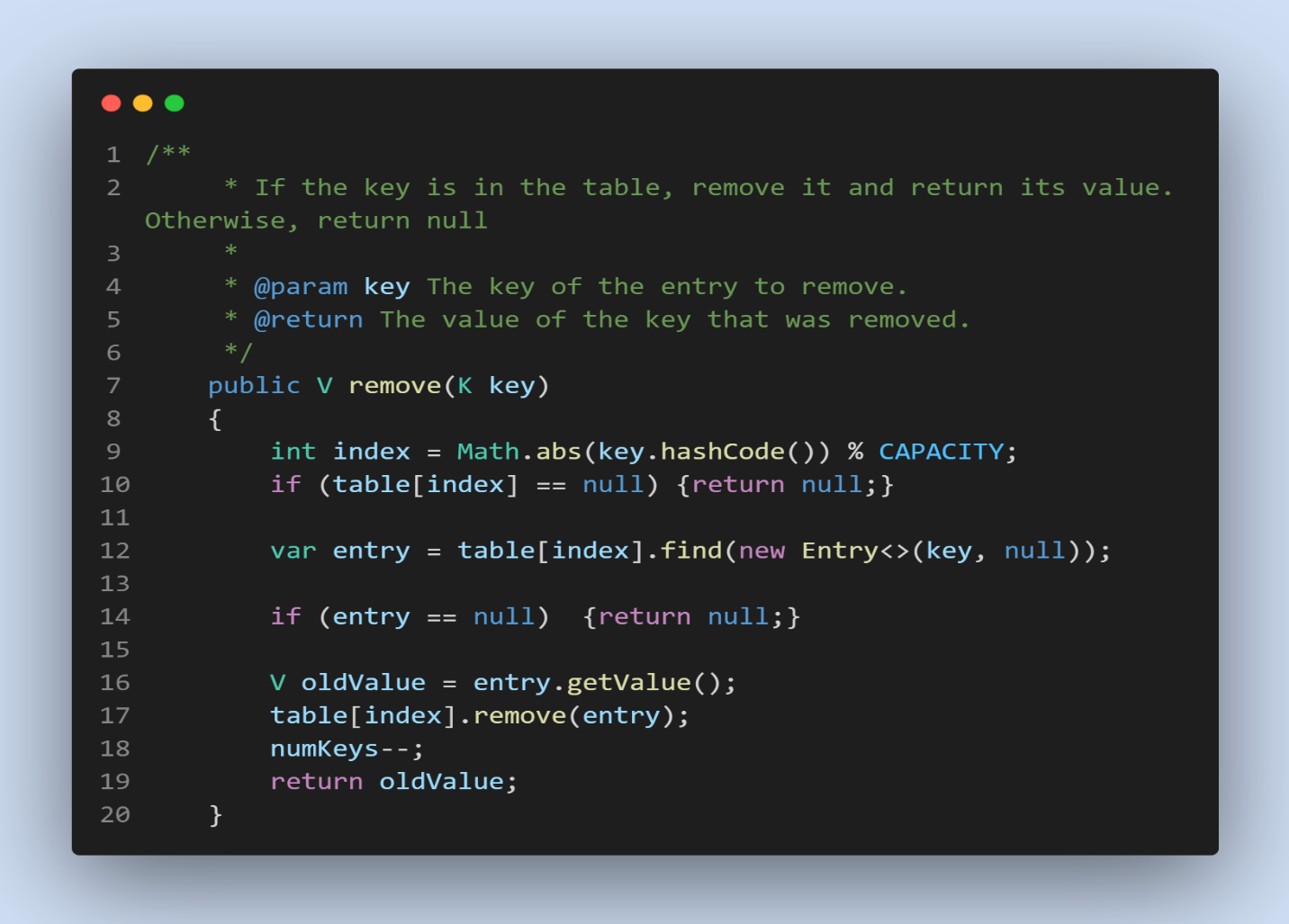


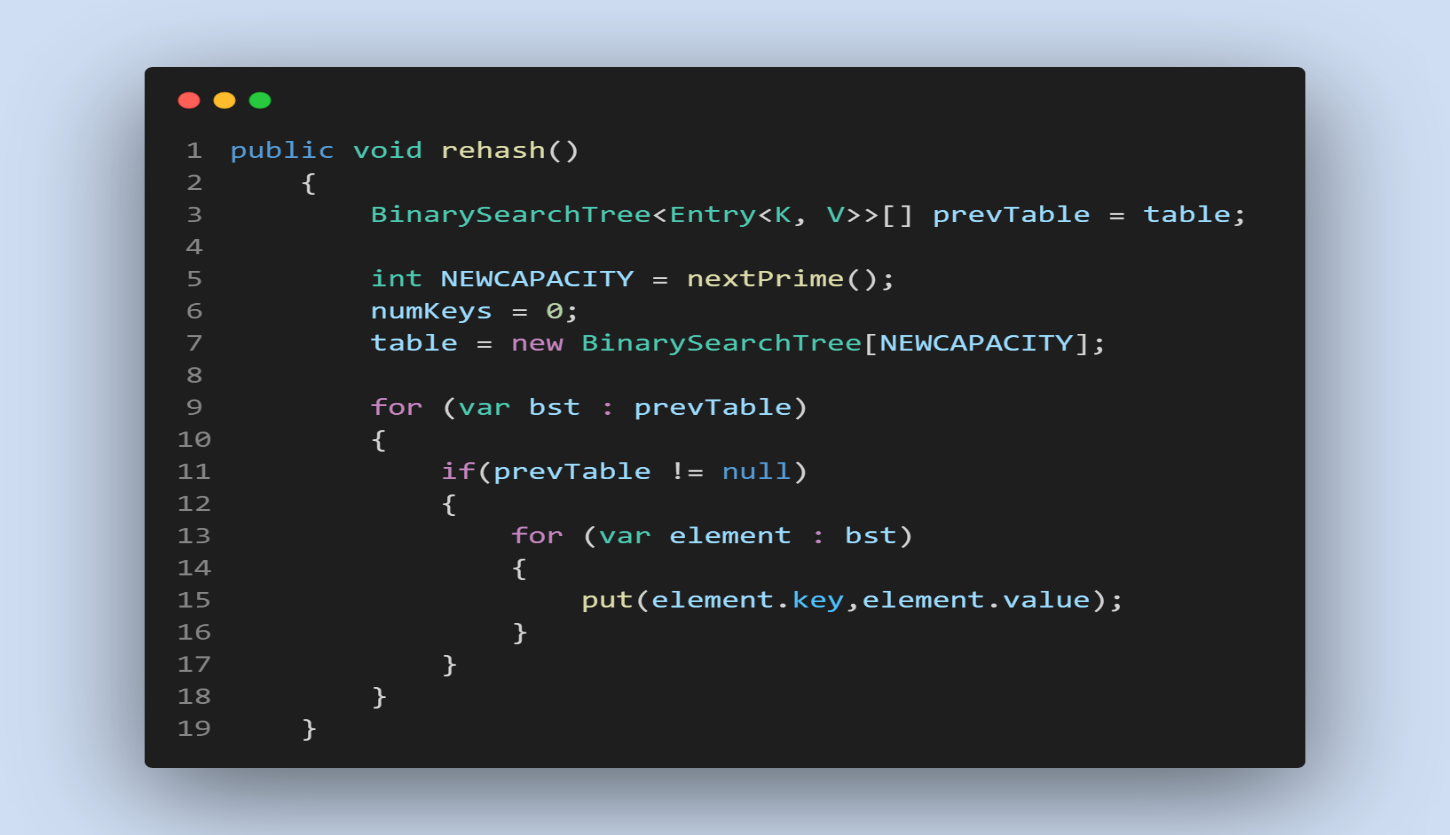
3 -) Compare 3 algorithms 100 times with 10000 sized arrays.

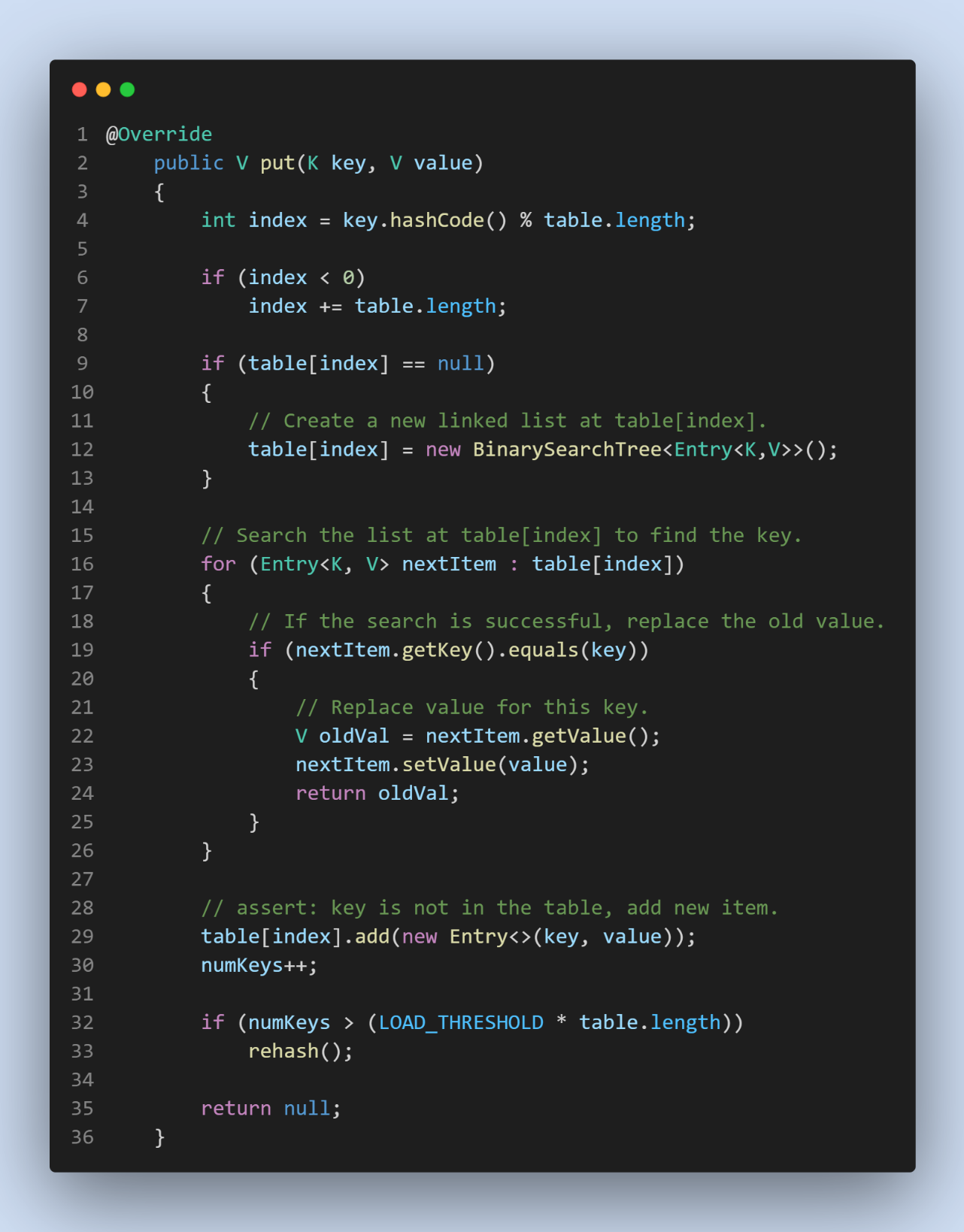


### 6 – Calculate Time complexity

--Example Hash Table functions implementations





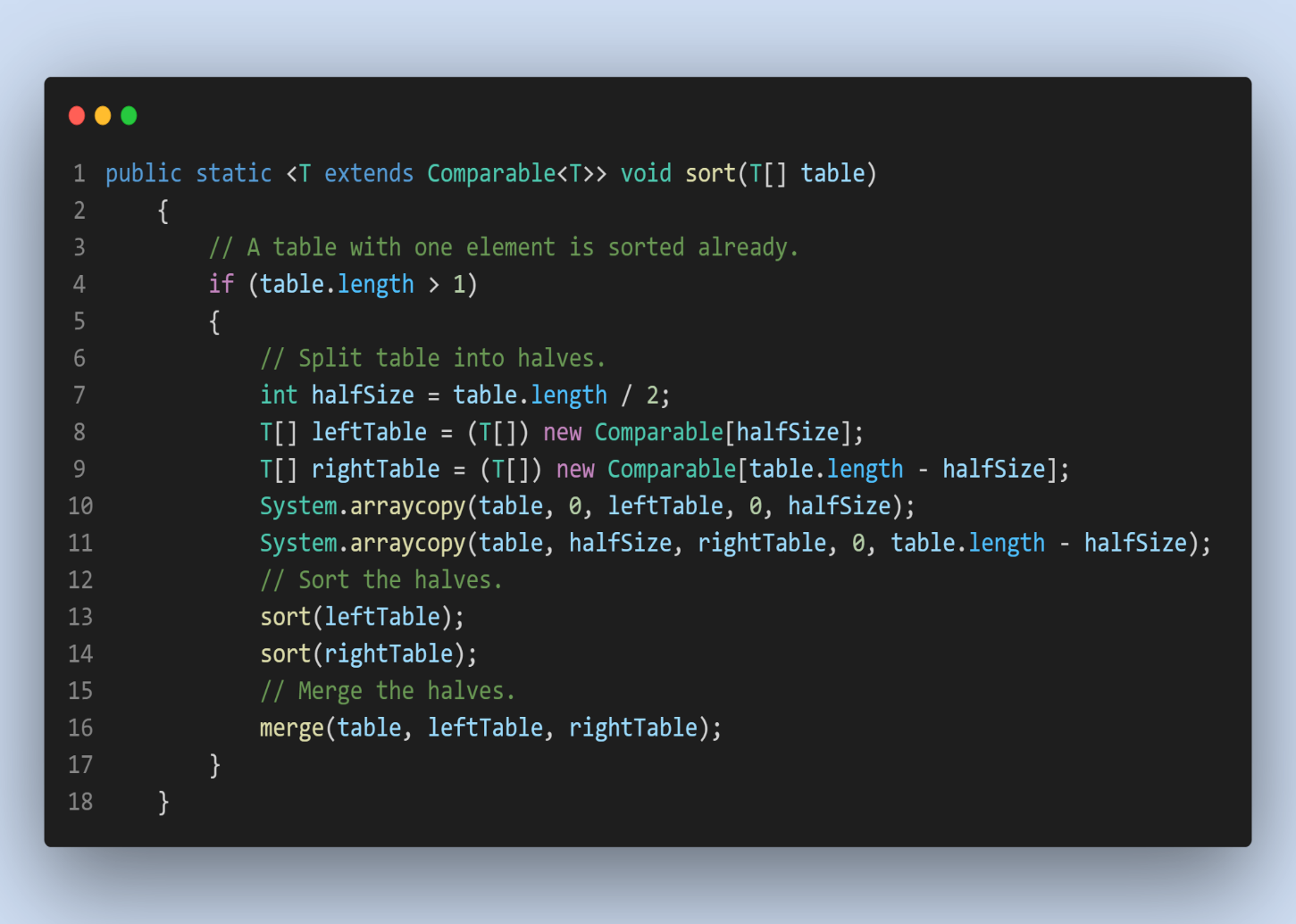


-- Hash table searching complexity average = O(logn).

-- Hash table searching complexity worst case = O(n).

-- Hash table searching complexity best case = O(1).

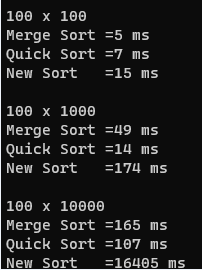
--Example Merge Sort functions implementations

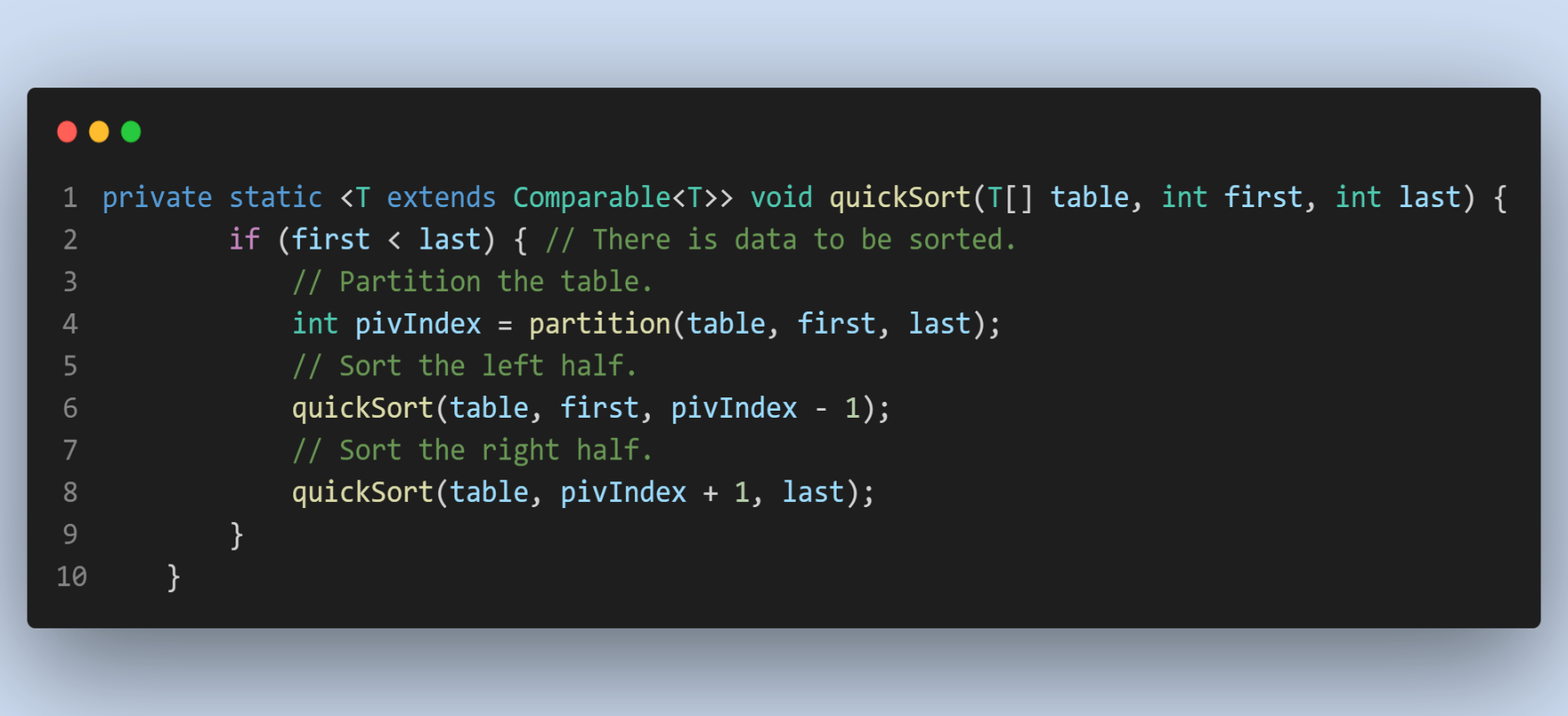


-- Merge Sort Average Complexity = O(nlogn)

-- Merge Sort Worst Case Complexity = O(nlogn)

-- Merge Sort Best Case Complexity = O(nlogn)

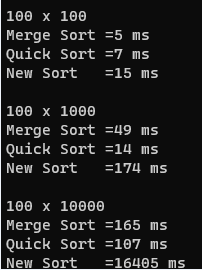


--Example Quick Sort functions implementations

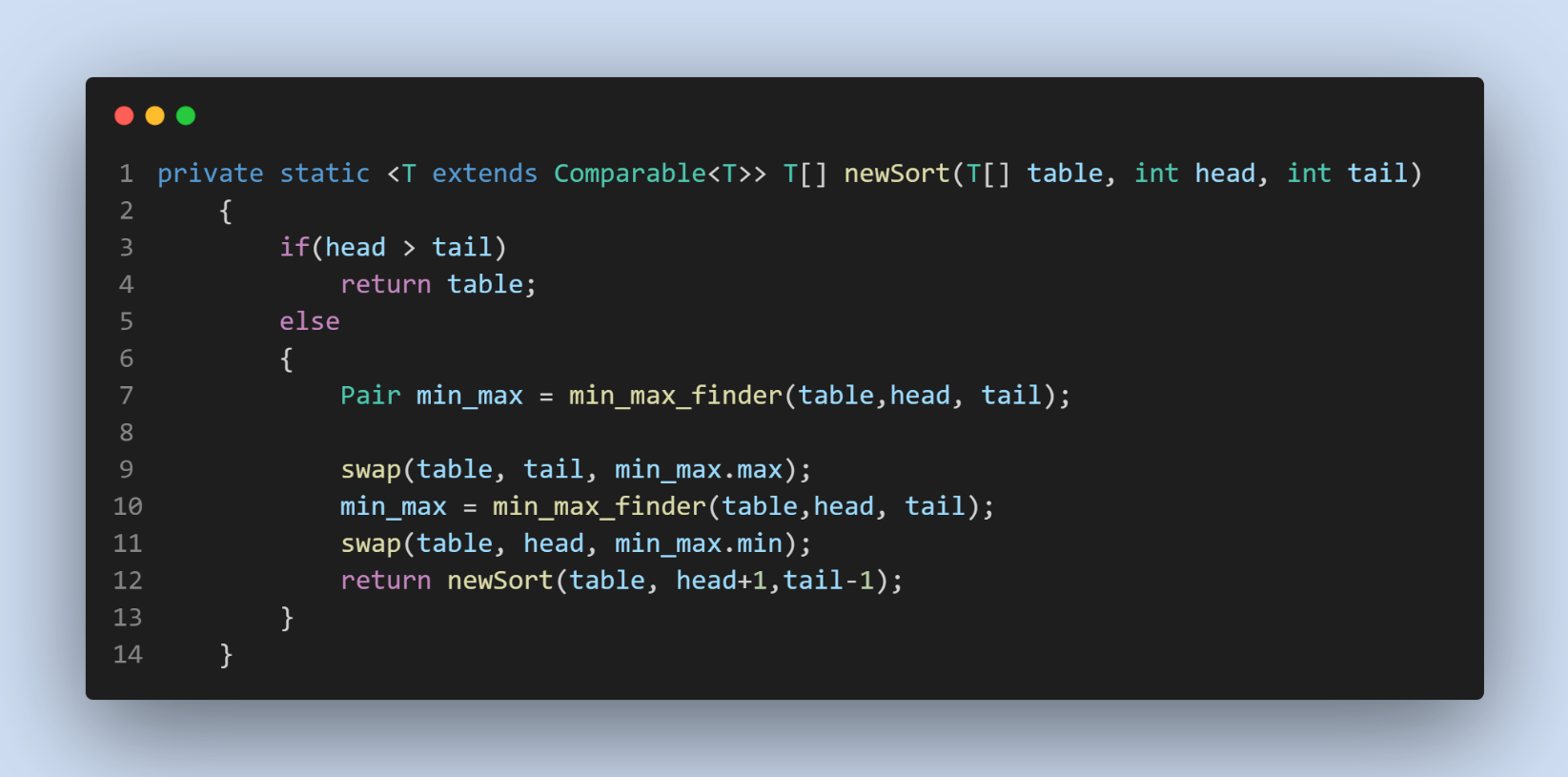
-- Quick Sort Average Complexity = O(nlogn)

-- Quick Sort Worst Case Complexity = O(n^2)

-- Quick Sort Best Case Complexity = O(nlogn)



-- Example Quick Sort functions implementations



-- New Sort Average Complexity = O(n^2)

-- New Sort Worst Case Complexity = O(n^2)

-- New Sort Best Case Complexity = O(n^2)

