Kaunas University of Technology

DATA STRUCTURES

Author: Abdelrhman Ibrahim

Group: IFU - 1

Teacher: Cenker Canbulut

Kaunas, 2022

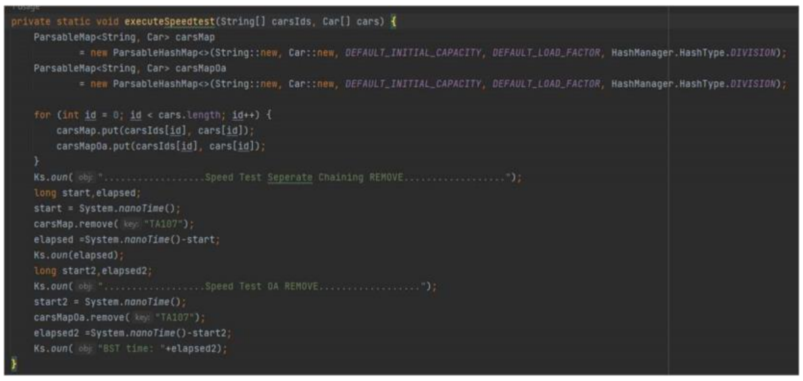
RESEARCH METHODS:

- remove(K key) – Removes the data point with the given key such as TA###;
- containsValue(Object value) - checks if one or more keys for the value specified by the method argument exist in the hash table;
- replace(K key, V oldValue, V newValue) – Finds the key and changes oldValue with the newValue. If key doesn‘t exist in the table, old value stays and method returns false.

COMPLEXITY

The worst time complexity of primary methods such as remove,insert and search are O(1).

SPEED TESTING METHOD



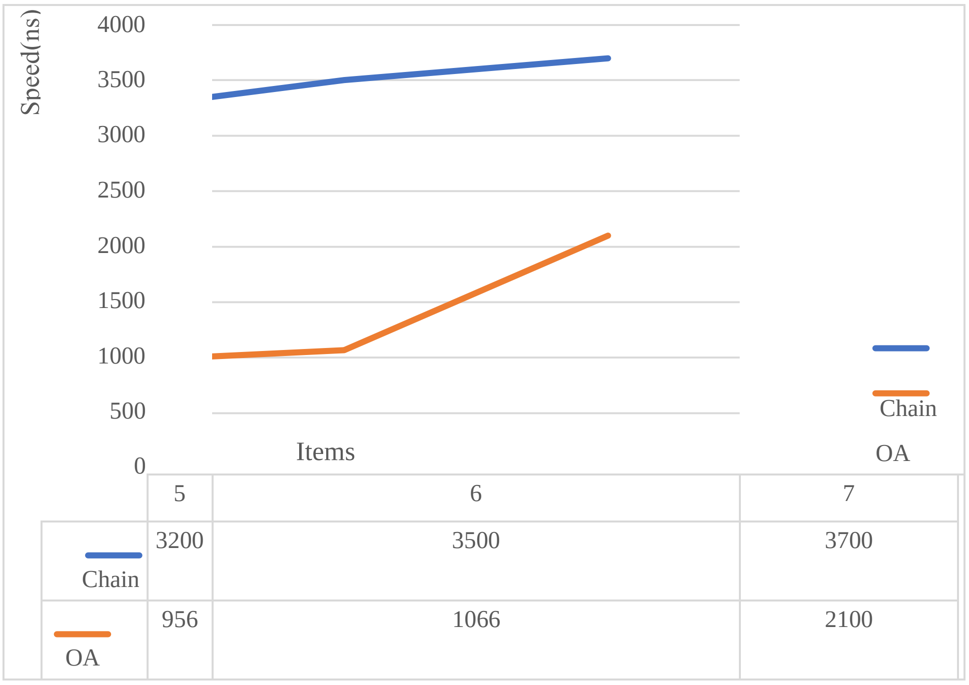
```java
private static void executeSpeedtest(String[] carsIds, Car[] cars) {
    ParsableMap<String, Car> carsMap
            = new ParsableHashMap<>(String::new, Car::new, DEFAULT_INITIAL_CAPACITY, DEFAULT_LOAD_FACTOR, HashManager.HashType.DIVISION);
    ParsableMap<String, Car> carsMapOa
            = new ParsableHashMap<>(String::new, Car::new, DEFAULT_INITIAL_CAPACITY, DEFAULT_LOAD_FACTOR, HashManager.HashType.DIVISION);

    for (int id = 0; id < cars.length; id++) {
        carsMap.put(carsIds[id], cars[id]);
        carsMapOa.put(carsIds[id], cars[id]);
    }
    Ks.oun( obj: "..................Speed Test Seperate Chaining REMOVE..................");
    long start,elapsed;
    start = System.nanoTime();
    carsMap.remove( key: "TA107");
    elapsed =System.nanoTime()-start;
    Ks.oun(elapsed);
    long start2,elapsed2;
    Ks.oun( obj: "..................Speed Test OA REMOVE..................");
    start2 = System.nanoTime();
    carsMapOa.remove( key: "TA107");
    elapsed2 =System.nanoTime()-start2;
    Ks.oun( obj: "BST time: "+elapsed2);
}
```

The testing method used here is simple. It is by starting time before the execution of the method and calculating the elapsed time as shown in the figure above and taking average of three runtimes.

BASIC PARAMETERS OF COMPUTER USED

PROCESSOR: intel Core i7 4th generation MEMORY: 16GB ddr3l RAM
SSD: 512GB

GRAPH



	5	6	7
Chain	3200	3500	3700
OA	956	1066	2100

CONCLUSION

The execution time here depends on the input size. We can easily calculate the running time using the order of increase. The remove method of class Open Addressing performs faster when compared to Seperate Chaining but for a more consistent performance seperate chaining can be used.