Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 5 REPORT

CELAL CAN KAYA
161044014

Course Assistant:Fatma Nur Esirci

1 Double Hashing Map

1.1 Pseudocode and Explanation

Put Metodu

1- Key veya value, null ise null return et.
2- Eğer size, capacity'nin 0.6 katına eşit veya fazla ise LoadBalancer devreye girsin.
3- 1. Hash Fonksiyonuyla indexi bul.
4- İndex 0'dan küçükse capacity ile topla.
5- Eğer bulunan index null ise elemanı yerleştir ve null return et.
6- Eğer bulunan indexteki keyler eşleşiyorsa, valueyu değiştir. Eski valueyi return et.
7- İndex null olana kadar veya indexteki key ile verilen key eşleşene kadar
    a. Double hashing ile index hesapla
    b. İndex 0'dan küçükse capacity ile topla
8- Eğer bulunan index null ise elemanı yerleştir ve null return et.
9- Eğer bulunan indexteki keyler eşleşiyorsa, valueyu değiştir. Eski valueyi return et.
10- Null Return et.

Get Metodu

1- Key veya value, null ise null return et.
2- 1. Hash Fonksiyonuyla indexi bul.
3- İndex 0'dan küçükse capacity ile topla.
4- Eğer bulunan indexteki keyler eşleşiyorsa, value'i return et.
5- İndex null olana kadar
    a. Double hashing ile index hesapla
    b. İndex 0'dan küçükse capacity ile topla
    c. İndex null değilse ve keyler eşleşiyorsa valueyi return et
6- Null Return et.

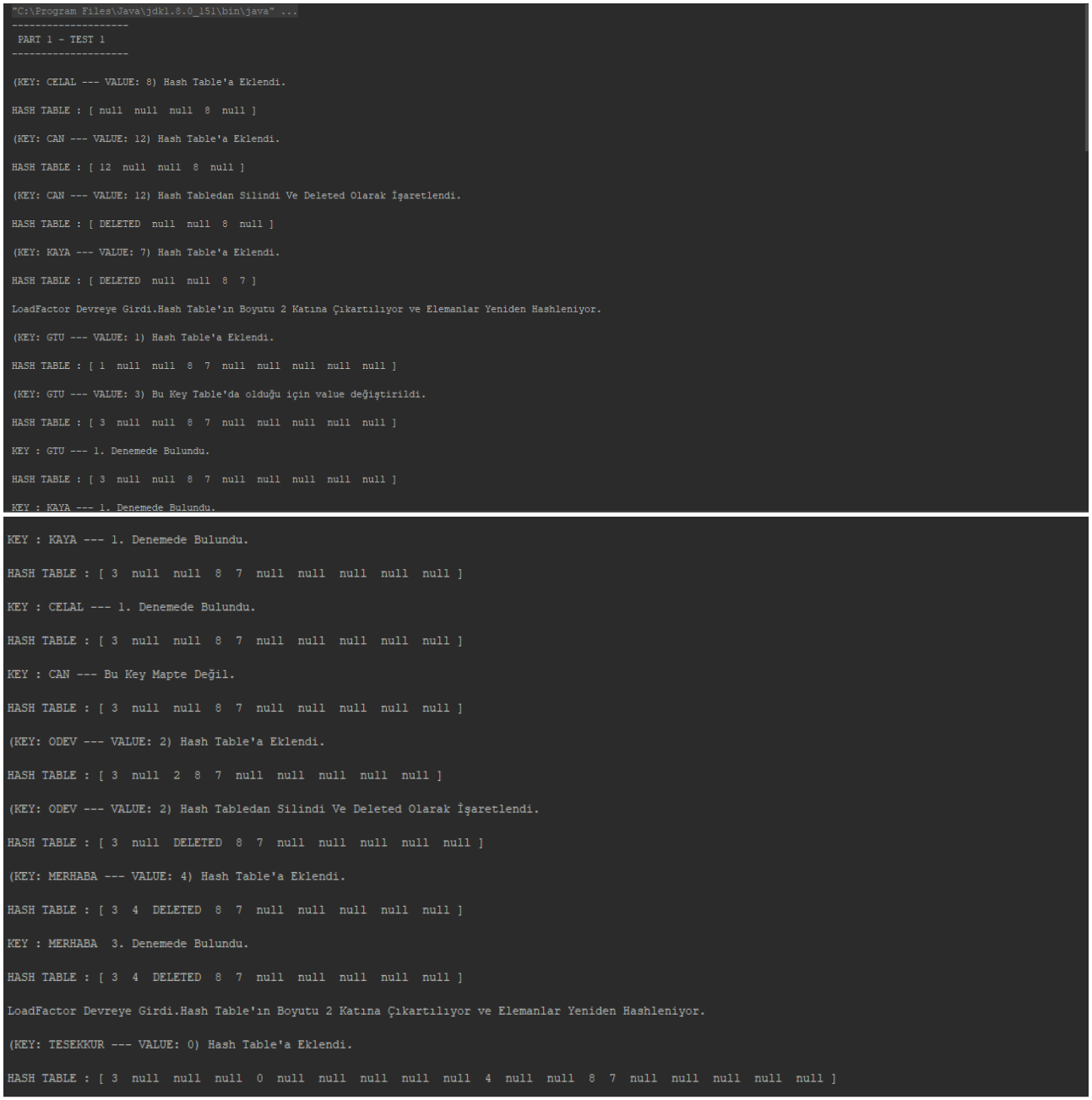
Add Metodu

1- Key veya value, null ise null return et.
2- 1. Hash Fonksiyonuyla indexi bul.
3- İndex 0'dan küçükse capacity ile topla.
4- Eğer bulunan indexteki keyler eşleşiyorsa, key ve value'yi null yap ve "Deleted" olarak işaretle.
5- İndex null olana kadar
    a. Double hashing ile index hesapla
    b. İndex 0'dan küçükse capacity ile topla
    c. İndex null değilse ve keyler eşleşiyorsa value'yi null yap ve "Deleted" olarak işaretle.
6- Null Return et.

Map Interfacesi implement Edildi. Hash1 ve Hash2 Metodları Yazıldı. Hash1 Metodu "Hashcode % Size" olarak belirlendi. Hash2 Metodu " (Prime – (HashCode % Prime)) Olarak belirlendi.Hash1 Metodu kullanılıp collision meydana geldiğinde Double Hashing ile yeni index bulundu. Hash Table'ın size'ının 0.6'sı veya fazlası dolduğunda LoadFactor devreye girerek Hash Table boyutu 2 katına çıkartıldı ve bütün elemanlar tekrardan hashlendi.

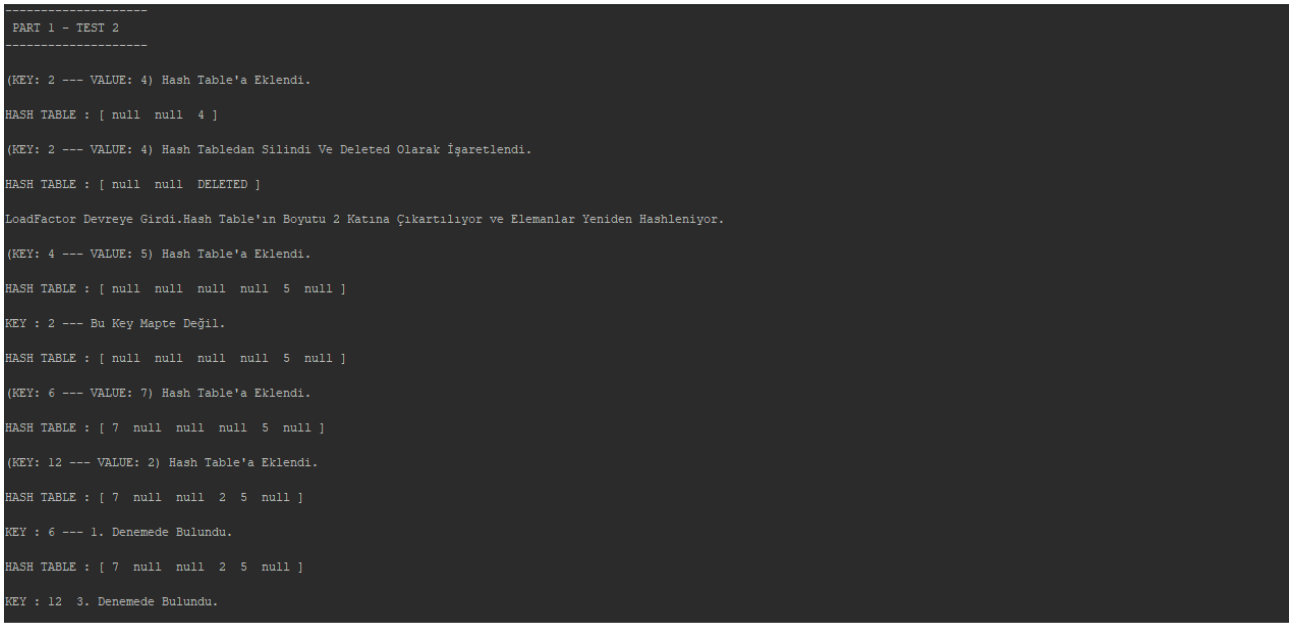
1.2 Test Cases

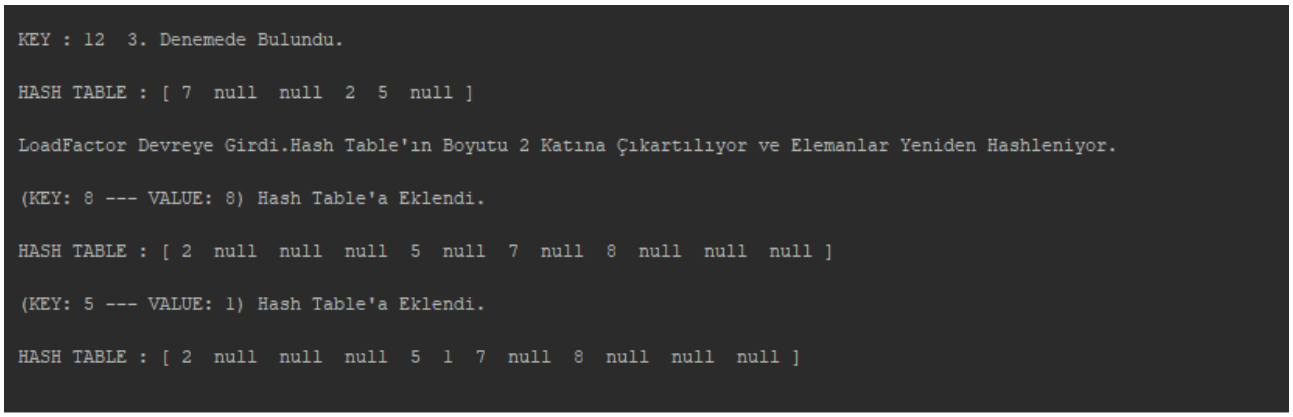
TEST 1



Resimlerde gerekli açıklamalar, yapılan işlemler belirtilmiştir.

TEST 2





Resimlerde gerekli açıklamalar, yapılan işlemler belirtilmiştir.

2 Recursive Hashing Set

2.1 Pseudocode and Explanation

Put Metodu

1- 1. Hash Fonksiyonuyla indexi bul.
2- Eğer bulunan index null ise elemanı yerleştir ve true return et.
3- Eğer bulunan indexteki element, ekleyeceğimiz elemana eşitse false return et.
4- Eğer indexteki eleman null değilse ve element, ekleyeceğimiz elemana eşit değilse ve index'in içindeki recursive array null ise yeni size'ı bul ve arrayi initialize et.Yeni arrayle metodu recursive olarak çağır.
5- Eğer indexteki eleman null değilse ve element, ekleyeceğimiz elemana eşit değilse recursive olan arrayle metodu recursive olarak çağır.
6- False return et.

Remove Metodu

1- 1. Hash Fonksiyonuyla indexi bul.
2- Eğer bulunan index null ise false return et.
3- Eğer bulunan indexteki element, sileceğimiz elemana eşitse elemanı sil, “Deleted” olarak işaretle ve size’ı azalt.
4- Eğer indexteki eleman null değilse ve element, sileceğimiz elemana eşit değilse ve index’in içindeki recursive array null ise yeni size’ı bul ve arrayi initialize et.Yeni arrayle metodu recursive olarak çağır.
5- Eğer indexteki eleman null değilse ve element, sileceğimiz elemana eşit değilse recursive olan arrayle metodu recursive olarak çağır.
6- False return et.

Set Interfacesi implement Edildi. Hash Metodu Yazıldı. Hash Metodu “Hashcode % Size” olarak belirlendi.Hash Metodu kullanılıp collision meydana geldiğinde Entry’min içindeki yeni arrayi sizedan küçük en büyük asal sayı ile initialize ediyorum ve Hash Metodunu yeni size’a göre güncelleyip tekrardan kontrol ediyorum.Collision meydana geldikçe yukardaki adım recursive olarak devam ediyor.

2.2 Test Cases

Bir elemanın yanında açılan parantez onun içindeki arrayi gösterecek şekilde ekrana bastırdım.

TEST 1

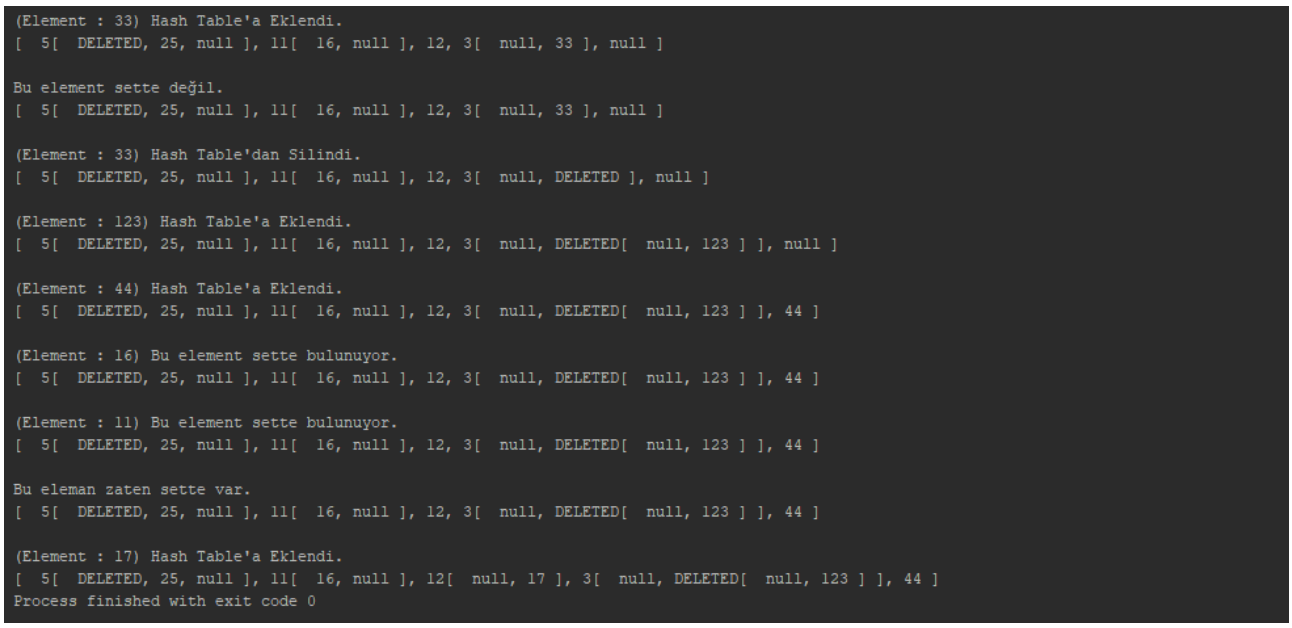
```
--------------------
 PART 2 - TEST
--------------------


(Element : 5) Hash Table'a Eklendi.
[  null, null, null, null, null, 5, null, null ]

(Element : 3) Hash Table'a Eklendi.
[  null, null, null, 3, null, 5, null, null ]

(Element : 2) Hash Table'a Eklendi.
[  null, null, 2, 3, null, 5, null, null ]

(Element : 1) Hash Table'a Eklendi.
[  null, 1, 2, 3, null, 5, null, null ]

(Element : 13) Hash Table'a Eklendi.
[  null, 1, 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 9) Hash Table'a Eklendi.
[  null, 1[  null, null, null, null, 9 ], 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 17) Hash Table'a Eklendi.
[  null, 1[  null, null, 17, null, 9 ], 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 25) Hash Table'a Eklendi.
[  null, 1[  25, null, 17, null, 9 ], 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 33) Hash Table'a Eklendi.
[  null, 1[  25, null, 17, 33, 9 ], 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 49) Hash Table'a Eklendi.
[  null, 1[  25, null, 17, 33, 9[  null, 49, null ] ], 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]
```

```
(Element : 49) Hash Table'a Eklendi.
[  null, 1[  25, null, 17, 33, 9[  null, 49, null ] ], 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 96) Hash Table'a Eklendi.
[  96, 1[  25, null, 17, 33, 9[  null, 49, null ] ], 2, 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 98) Hash Table'a Eklendi.
[  96, 1[  25, null, 17, 33, 9[  null, 49, null ] ], 2[  98, null ], 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 97) Hash Table'a Eklendi.
[  96, 1[  25, null, 17[  null, 97 ], 33, 9[  null, 49, null ] ], 2[  98, null ], 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

Bu element sette değil.
[  96, 1[  25, null, 17[  null, 97 ], 33, 9[  null, 49, null ] ], 2[  98, null ], 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

Bu element sette değil.
[  96, 1[  25, null, 17[  null, 97 ], 33, 9[  null, 49, null ] ], 2[  98, null ], 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 17) Hash Table'dan Silindi.
[  96, 1[  25, null, DELETED[  null, 97 ], 33, 9[  null, 49, null ] ], 2[  98, null ], 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 49) Hash Table'dan Silindi.
[  96, 1[  25, null, DELETED[  null, 97 ], 33, 9[  null, DELETED, null ] ], 2[  98, null ], 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]

(Element : 2) Bu element sette bulunuyor.
[  96, 1[  25, null, DELETED[  null, 97 ], 33, 9[  null, DELETED, null ] ], 2[  98, null ], 3, null, 5[  null, null, null, null, null, null, 13 ], null, null ]
Process finished with exit code 0
```

Resimlerde gerekli açıklamalar, yapılan işlemler belirtilmiştir.

TEST 2

```
--------------------
 PART 2 - TEST 2
--------------------


(Element : 5) Hash Table'a Eklendi.
[  5, null, null, null, null ]

(Element : 3) Hash Table'a Eklendi.
[  5, null, null, 3, null ]

(Element : 11) Hash Table'a Eklendi.
[  5, 11, null, 3, null ]

(Element : 0) Hash Table'a Eklendi.
[  5[  0, null, null ], 11, null, 3, null ]

(Element : 0) Hash Table'dan Silindi.
[  5[  DELETED, null, null ], 11, null, 3, null ]

(Element : 12) Hash Table'a Eklendi.
[  5[  DELETED, null, null ], 11, 12, 3, null ]

(Element : 16) Hash Table'a Eklendi.
[  5[  DELETED, null, null ], 11[  16, null ], 12, 3, null ]

(Element : 25) Hash Table'a Eklendi.
[  5[  DELETED, 25, null ], 11[  16, null ], 12, 3, null ]

(Element : 33) Hash Table'a Eklendi.
[  5[  DELETED, 25, null ], 11[  16, null ], 12, 3[  null, 33 ], null ]

Bu element sette değil.
[  5[  DELETED, 25, null ], 11[  16, null ], 12, 3[  null, 33 ], null ]
```



Resimlerde gerekli açıklamalar, yapılan işlemler belirtilmiştir.

3 Sorting Algortihms

3.1 MergeSort with DoubleLinkedList

This part about Question3 in HW5

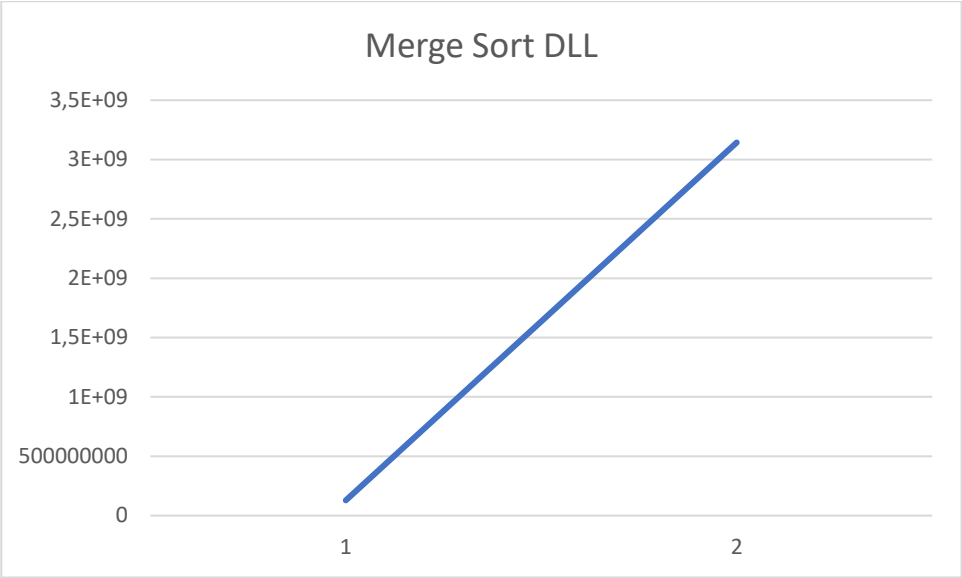
3.1.1 Pseudocode and Explanation

Pseudocode

1-) LinkedList'i 1 elemanı kalana kadar 2'ye böl
2-) Kalan 1 elemanlı 2 listeyi karşılaştır ve birleştir
    a-) Bu işlemi yeni oluşan listeyle recursive olarak devam ettir.

3.1.2 Average Run Time Analysis

Merge Sort'u çok verimsiz yazdığım için çalışma süresi çok uzun sürdü. Sadece ilk 2 size için hesapladım. Ayrıca bunu en sondaki karşılaştırma listesine diğerlerini daha ayrıntılı göstermek için dahil etmeyeceğim.



Grafikte 2 Size Olduğu için böyle saçma bir grafik ortaya çıktı.

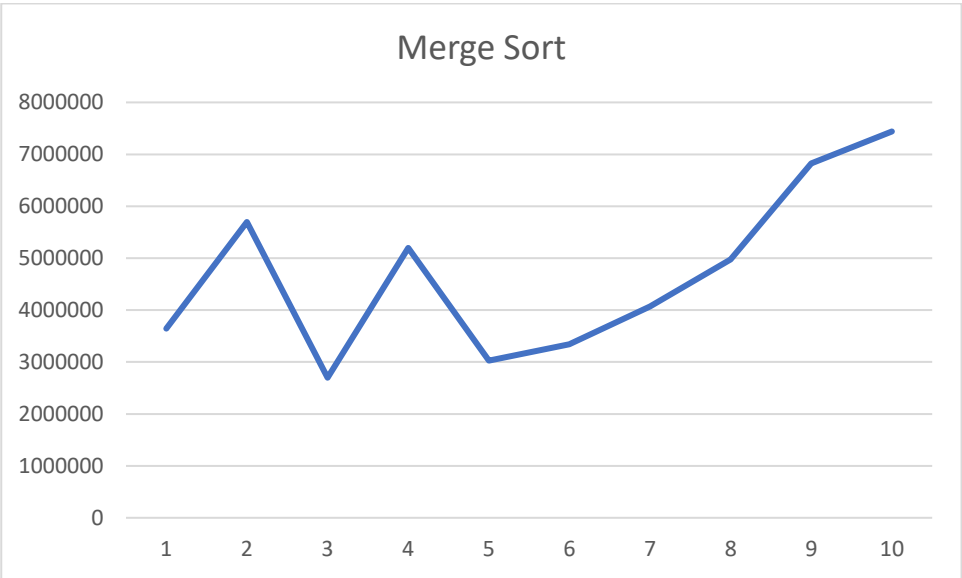
3.1.3 Wort-case Performance Analysis

Merge Sort'un Worst Case'iyle Average Case'i arasında fark olmayacaktır. Fakat yukardada belirttiğim gibi çok uzun süreler aldığından bu partı eklemeyeceğim.

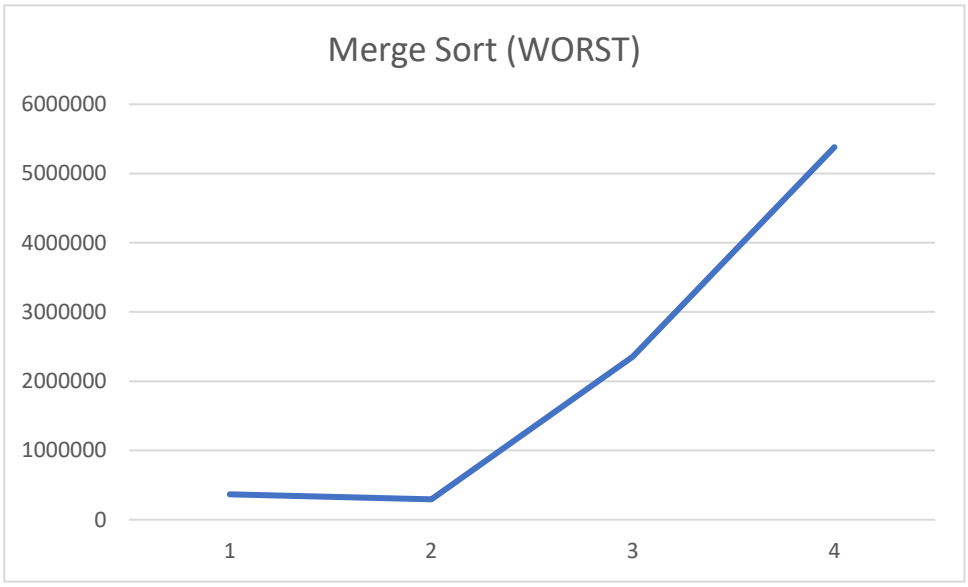
3.2 MergeSort

This part about code in course book.

3.2.1 Average Run Time Analysis

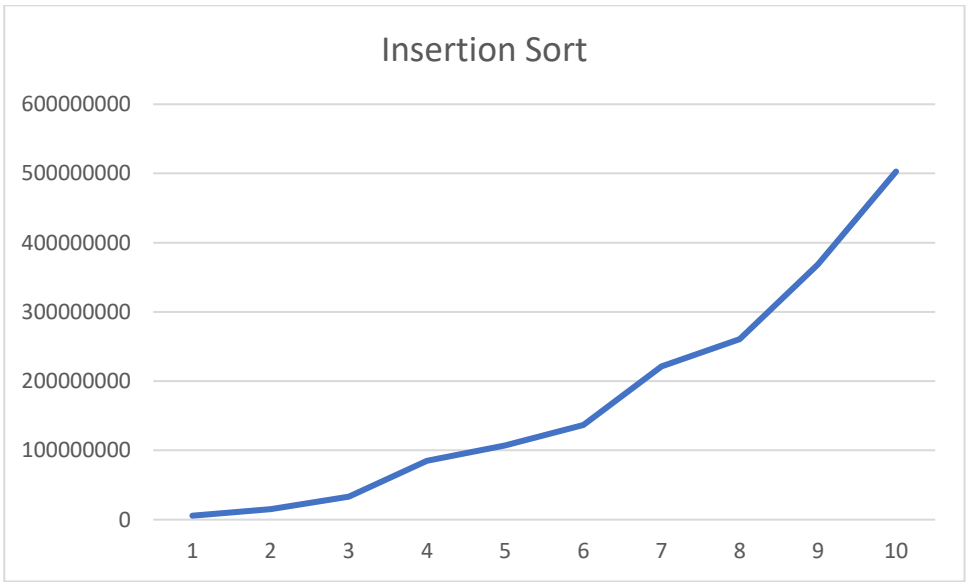


3.2.2 Wort-case Performance Analysis

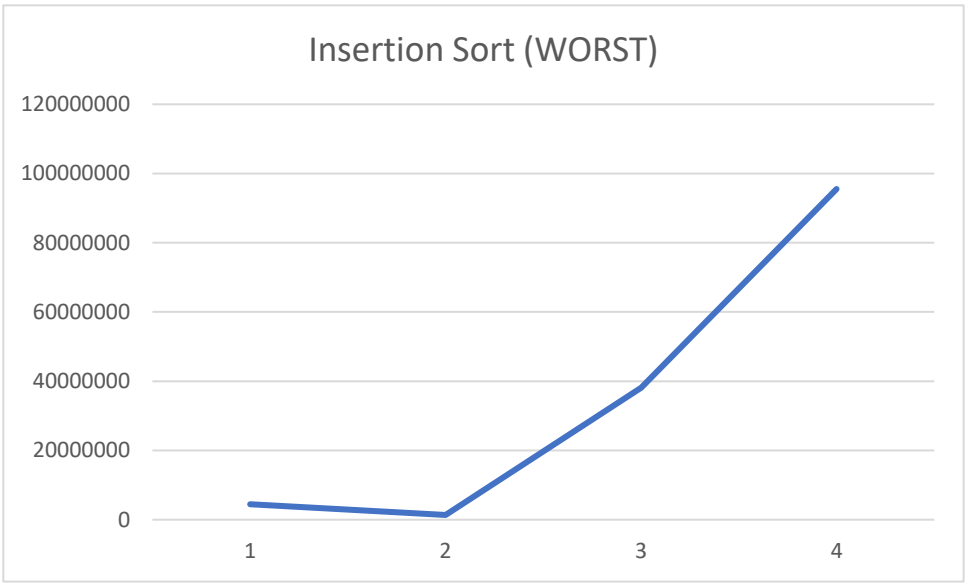


3.3 Insertion Sort

3.3.1 Average Run Time Analysis

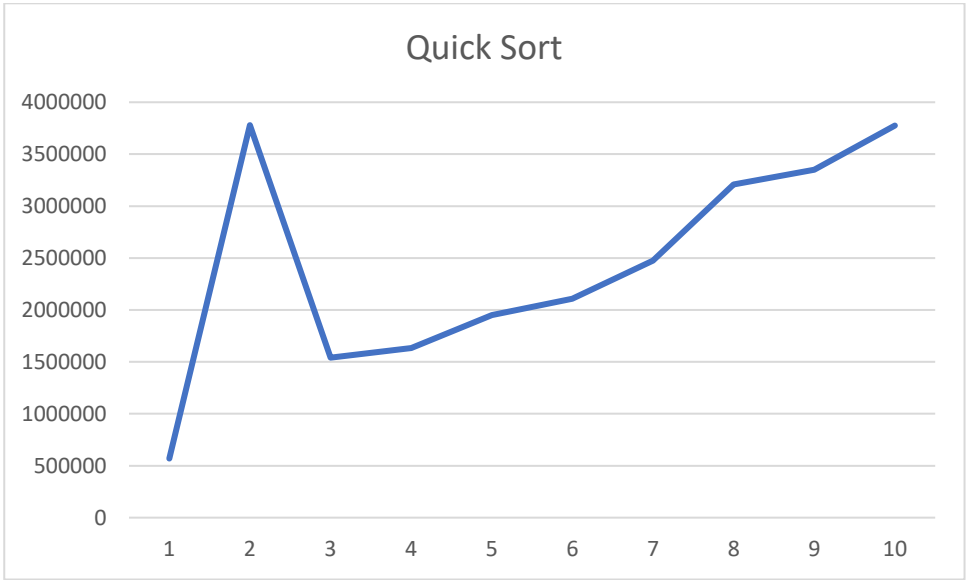


3.3.2 Wort-case Performance Analysis



3.4 Quick Sort

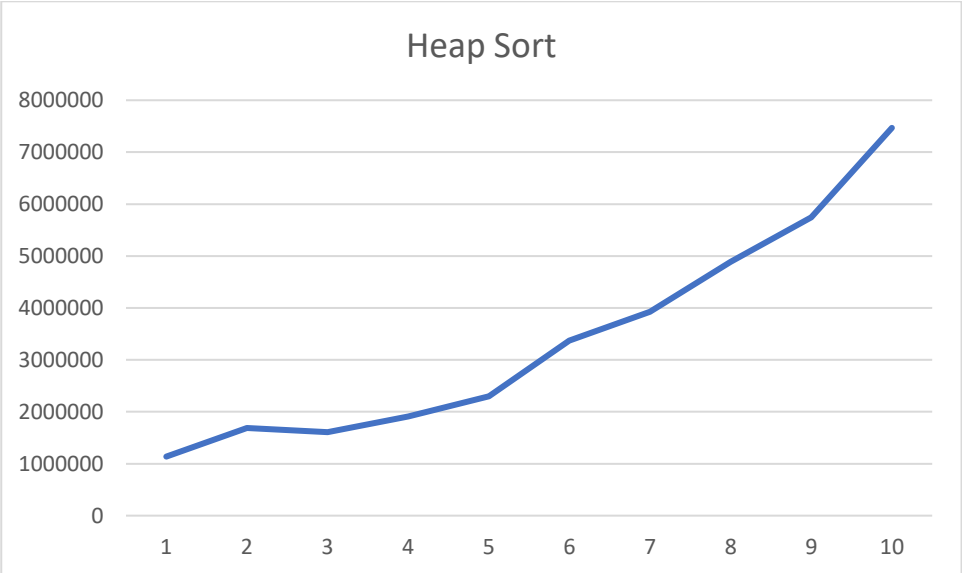
3.4.1 Average Run Time Analysis



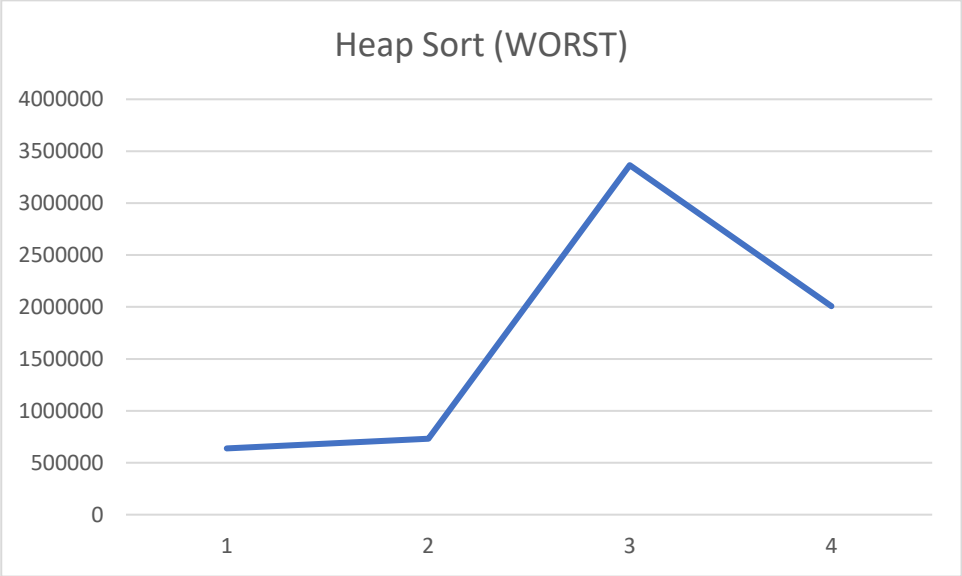
3.4.2 Wort-case Performance Analysis

3.5 Heap Sort

3.5.1 Average Run Time Analysis



3.5.2 Wort-case Performance Analysis



4 Comparison the Analysis Results