GTU Department of Computer Engineering

CSE 222- Spring 2020

Homework 7 Report

FATİH OĞUZ

151044025

AVL TREE

0: 20

null

null

****

1: 20

null

0: 30

null

null

****

0: 20

0: 8

null

null

0: 30

null

null

****

1: 20

0: 8

null

null

1: 30

null

0: 47

null

null

****

1: 20

0: 8

null

null

0: 39

0: 30

null

null

0: 47

null

null

****

0: 20

1: 8

null

0: 18

null

null

0: 39

0: 30

null

null

0: 47

null

null

```
**************************************************************************
1: 20
        1: 8
                null
                0: 18
                        null
                        null
        1: 39
                0: 30
                        null
                        null
                -1: 47
                        0: 40
                                null
                                null
                        null

**************************************************************************
1: 20
        1: 8
                null
                0: 18
                        null
                        null
        0: 39
                -1: 30
                        0: 25
                                null
```

null

null

-1: 47

0: 40

null

null

null

****

1: 18

0: 8

null

null

0: 39

-1: 30

0: 25

null

null

null

-1: 47

0: 40

null

null

null

****

1: 18

0: 8

null

null

1: 39

0: 25

null

null

-1: 47

0: 40

null

null

null

****

0: 39

0: 18

null

0: 25

null

null

-1: 47

0: 40

null

null

null

****

-1: 39

0: 18

null

0: 25

null

null

0: 40

null

null

****

-1: 25

-1: 18

null

null

0: 40

null

null

****

0: 25

null

0: 40

null

null

****

-1: 25

null

null

****

null

RED BLACK TREE

RED BLACK TREE

RULES:

- Every node is red or black
- Root is alvays black
- New insertions are always red
- Every path from root-leaf has the same number of black nodes
- No path can have two consecutive red nodes
- Null are black
-

Rebalance Black Aunt Rotate :BAR

Red Aunt COLORFLIP

AFTER ROTATION

BLACK

RED RED

AFTER COLORFLIP

RED

BLACK BLACK

```
Black: 20
        null
        null

********************************************************************************
Black: 20
        null
        Red    : 30
                 null
                 null

********************************************************************************
Black: 20
        Red    : 8
                 null
                 null
        Red    : 30
                 null
                 null

********************************************************************************
Black: 20
        Black: 8
                 null
                 null
        Black: 30
                 null
                 Red    : 47
                          null
```

null

****

Black: 20

Black: 8

null

null

Black: 39

Red : 30

null

null

Red : 47

null

null

****

Black: 20

Black: 8

null

Red : 18

null

null

Black: 39

Red : 30

null

null

Red : 47

null

null

```
**********************************************************************************
Black: 20
        Black: 8
                null
                Red    : 18
                        null
                        null
        Red    : 39
                Black: 30
                        null
                        null
                Black: 47
                        Red    : 40
                                null
                                null
                        null

**********************************************************************************
Black: 20
        Black: 8
                null
                Red    : 18
                        null
                        null
        Red    : 39
                Black: 30
                        Red    : 25
                                null
```

null

null

Black: 47

Red : 40

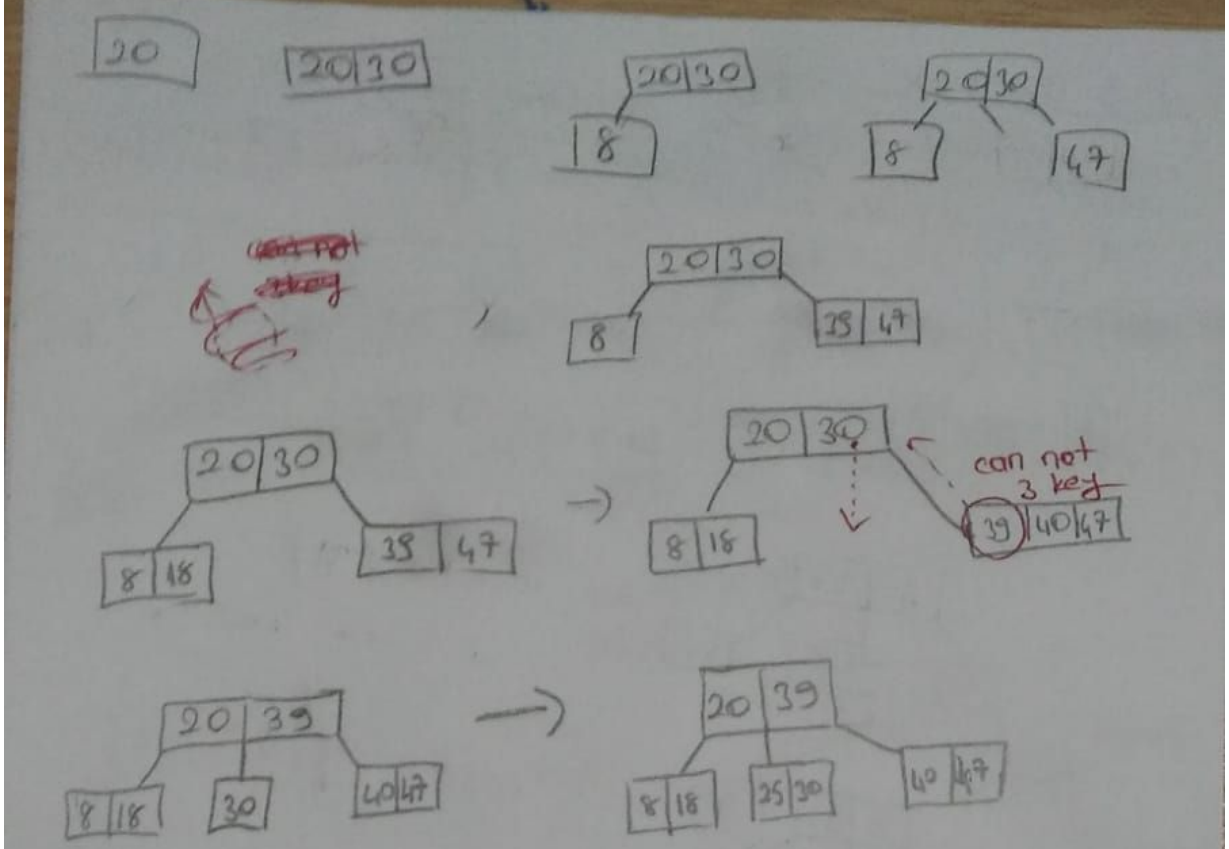
null

null

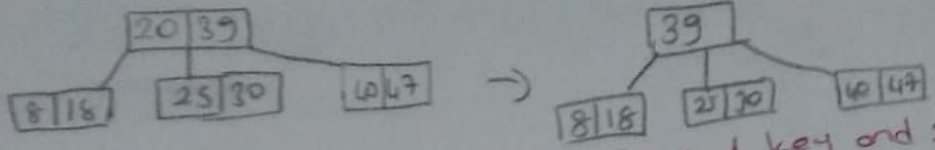
null

****

2 – 3 TREE



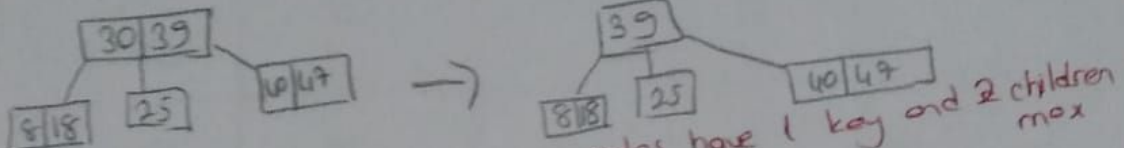
- Nodes have 1 ~~key~~ and 2 children max
- Nodes have 2 keys and 3 children max
- 2-3 trees are b-trees of order (at most 3 children on non-leaf nodes)



nodes have 1 key and 2 childsen max

40 → problem 25-30

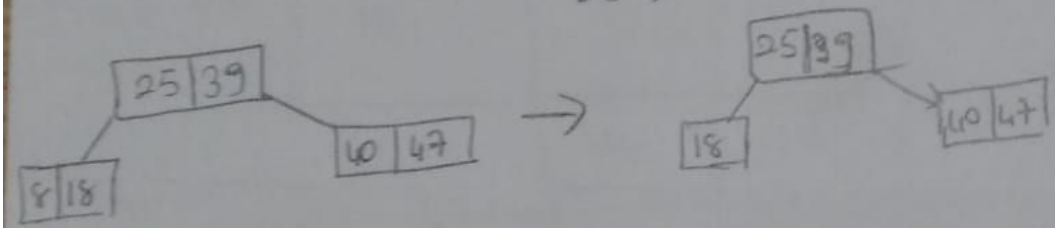
30 →

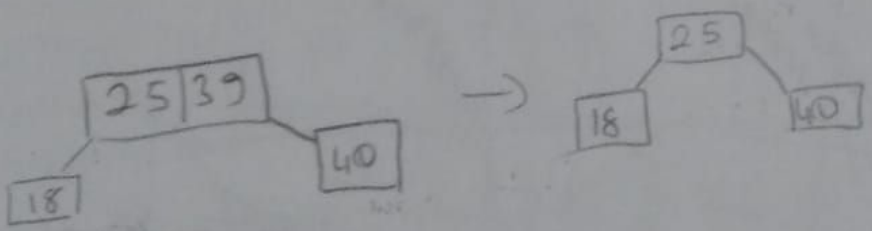


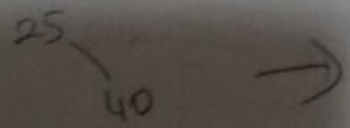
nodes have 1 key and 2 children max

40 → problem 25

25 →





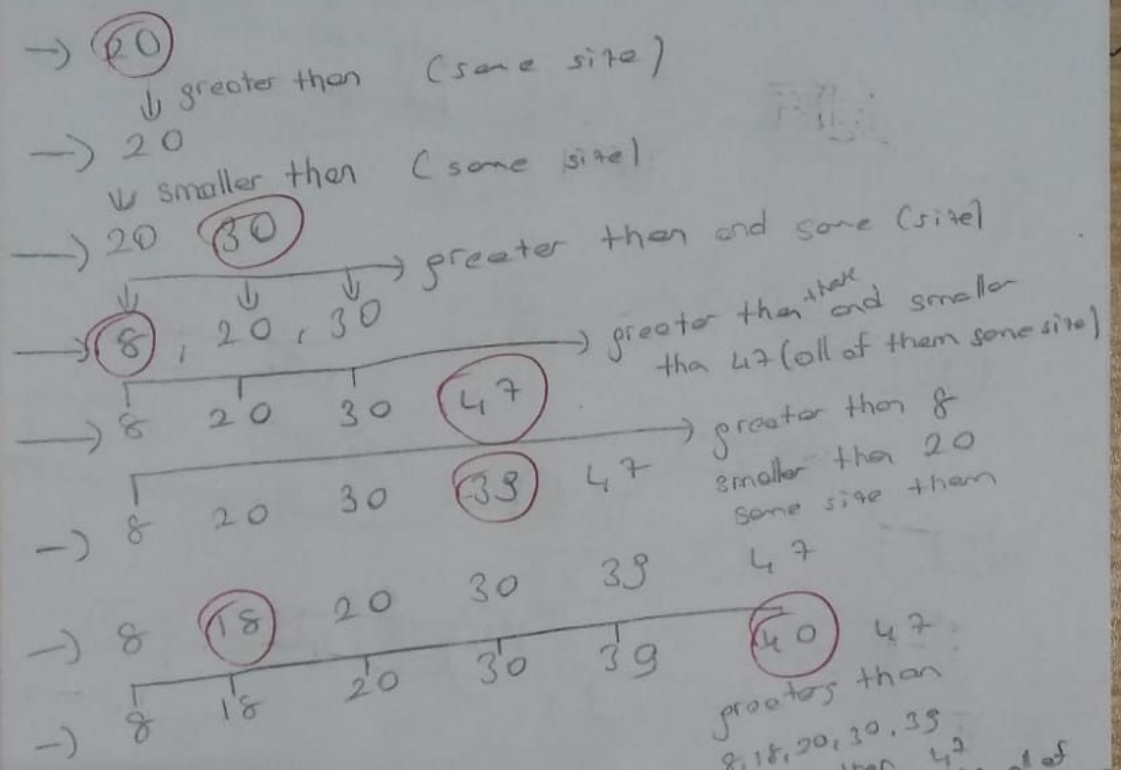


25 → null

SKIP LIST

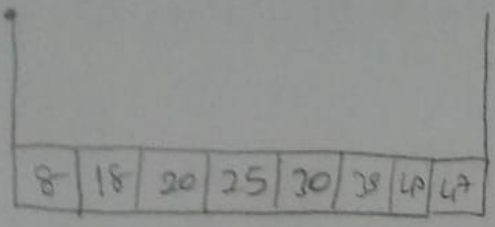
Skip List

20, 30, 8, 47, 39, 18, 40, 25



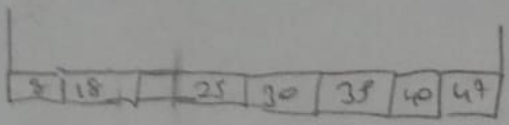
greaters than 8, 18, 20 and same size smaller than 30.

remove (20)



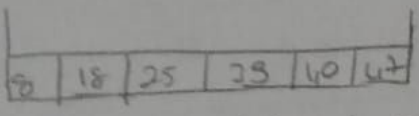
Search key (20) and remove all item (20)

remove (30)



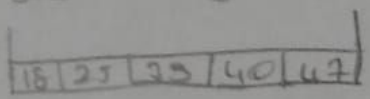
Search key (30) and remove all item (30)

remove (8)



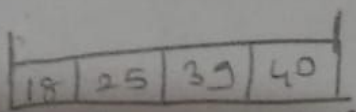
Search key (8) and remove all item (8)

remove (47)



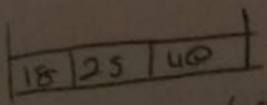
Search key (47) and remove all item (47)

remove (35)



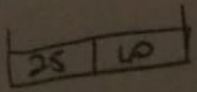
Search key (35) and remove all item (35)

remove (18)



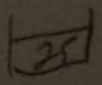
Search key (18) and remove all item (18)

remove (40)



Search key (40) and remove all item (40)

remove (25)

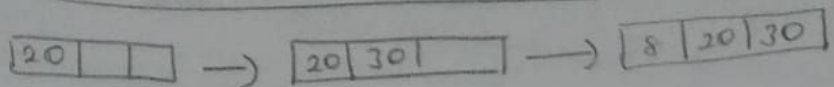


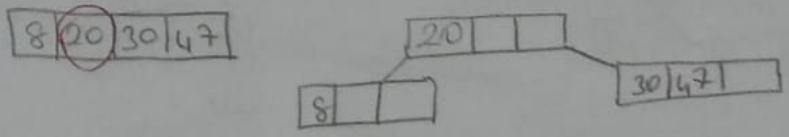
Search key (25) and remove all item (25)

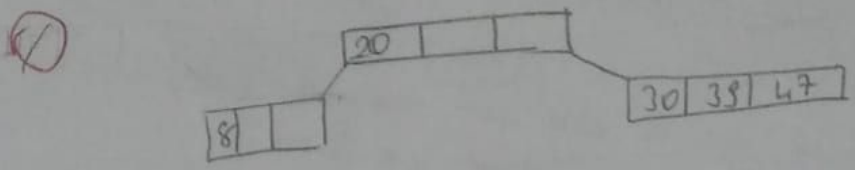
NULL

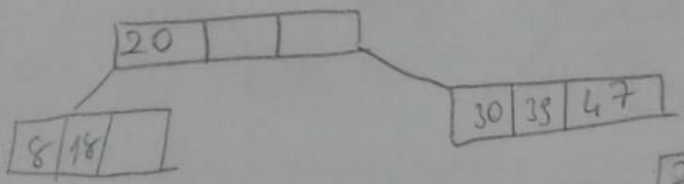
B TREE

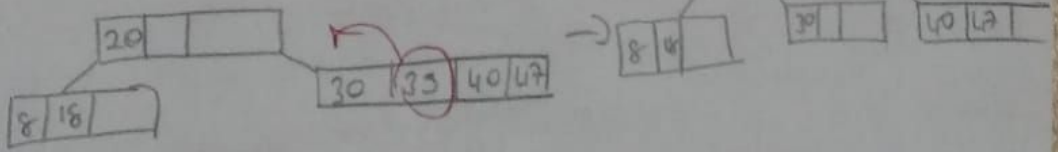
20, 30, 8, 47, 39, 18, 40, 25

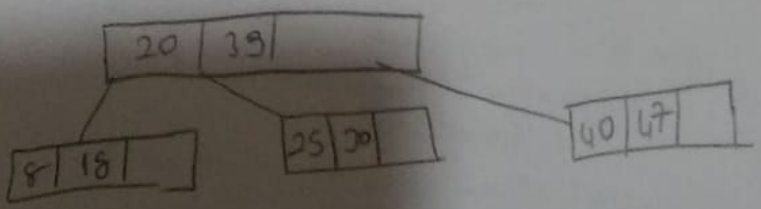




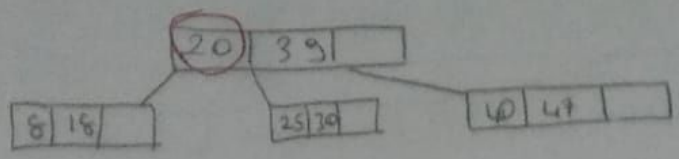




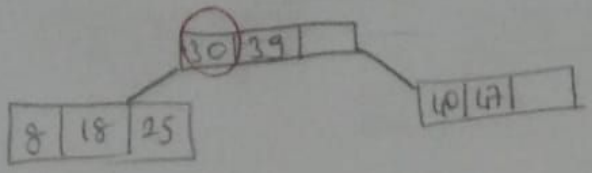


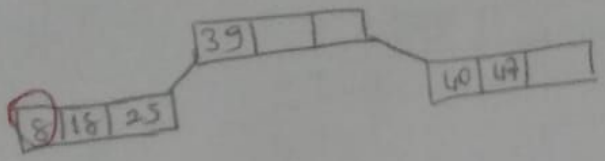


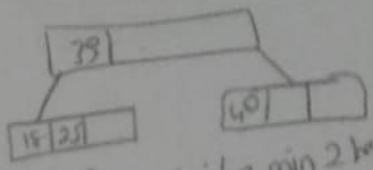
20, 30, 8, 47, 39, 18, 40, 25

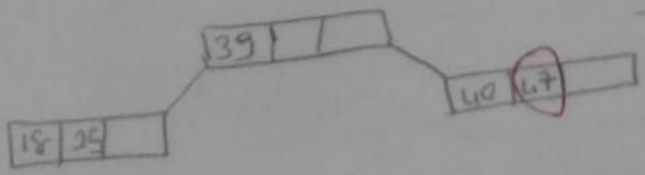


only root one keys
leaf must be minimum 2 keys

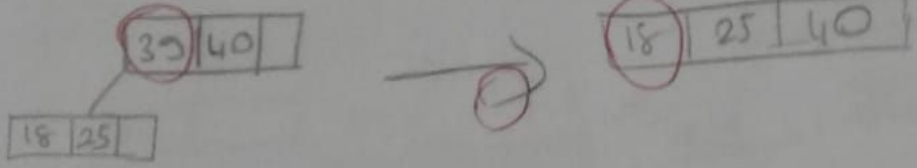


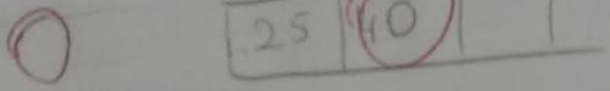


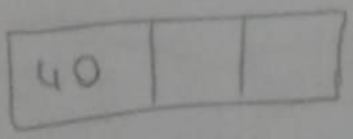


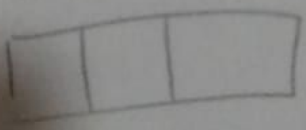


leaf must be min 2 keys

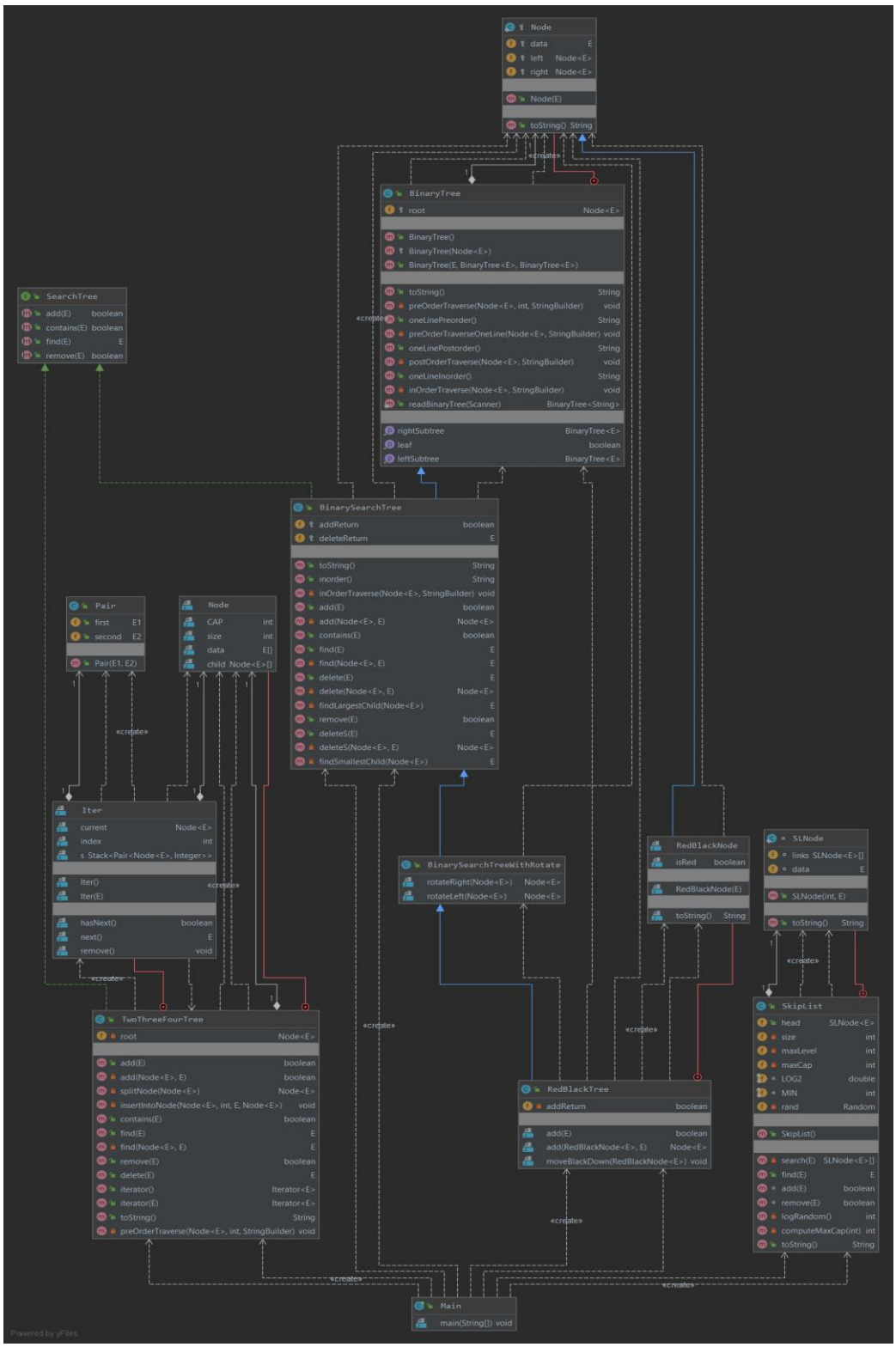








PART3


Node
data E
left Node<E>
right Node<E>
Node(E)
toString() String
«create»
BinaryTree
root Node<E>
BinaryTree()
BinaryTree(Node<E>)
BinaryTree(E, BinaryTree<E>, BinaryTree<E>)
toString() String
preOrderTraverse(Node<E>, int, StringBuilder) void
oneLinePreorder() String
preOrderTraverseOneLine(Node<E>, StringBuilder) void
oneLinePostorder() String
postOrderTraverse(Node<E>, StringBuilder) void
oneLineInorder() String
inOrderTraverse(Node<E>, StringBuilder) void
readBinaryTree(Scanner) BinaryTree<String>
rightSubtree BinaryTree<E>
leaf boolean
leftSubtree BinaryTree<E>
SearchTree
add(E) boolean
contains(E) boolean
find(E) E
remove(E) boolean
BinarySearchTree
addReturn boolean
deleteReturn E
toString() String
inorder() String
inOrderTraverse(Node<E>, StringBuilder) void
add(E) boolean
add(Node<E>, E) Node<E>
contains(E) boolean
find(E) E
find(Node<E>, E) E
delete(E) E
delete(Node<E>, E) Node<E>
findLargestChild(Node<E>) E
remove(E) boolean
deleteS(E) E
deleteS(Node<E>, E) Node<E>
findSmallestChild(Node<E>) E
Pair
first E1
second E2
Pair(E1, E2)
Node
CAP int
size int
data E[]
child Node<E>[]
«create»
Iter
current Node<E>
index int
s Stack<Pair<Node<E>, Integer>>
Iter()
Iter(E)
hasNext() boolean
next() E
remove() void
BinarySearchTreeWithRotate
rotateRight(Node<E>) Node<E>
rotateLeft(Node<E>) Node<E>
RedBlackNode
isRed boolean
RedBlackNode(E)
toString() String
SLNode
links SLNode<E>[]
data E
SLNode(int, E)
toString() String
TwoThreeFourTree
root Node<E>
add(E) boolean
add(Node<E>, E) boolean
splitNode(Node<E>) Node<E>
insertIntoNode(Node<E>, int, E, Node<E>) void
contains(E) boolean
find(E) E
find(Node<E>, E) E
remove(E) boolean
delete(E) E
iterator() Iterator<E>
iterator(E) Iterator<E>
toString() String
preOrderTraverse(Node<E>, int, StringBuilder) void
RedBlackTree
addReturn boolean
add(E) boolean
add(RedBlackNode<E>, E) Node<E>
moveBlackDown(RedBlackNode<E>) void
SkipList
head SLNode<E>
size int
maxLevel int
maxCap int
LOG2 double
MIN int
rand Random
SkipList()
search(E) SLNode<E>[]
find(E) E
add(E) boolean
remove(E) boolean
logRandom() int
computeMaxCap(int) int
toString() String
Main
main(String[]) void
Powered by yFiles

Problem Solutions Approach

After creating my data structures in the desired amount I added randomly.

I kept the random number generation limit at 100. I made them from 1000 to 1010 to make sure there were subsequent additions and removals. I found the working times in nanoseconds for each.

After running the program 10 times and getting the results in an excell file, I calculated their average.

I made a table of the average values and convert grafic them.

Running commands and Results

10 element addition times in nanoseconds

binarySearchTree1 51400

redBlackTreeBook1 49500

redBlackTreeJava1 58000

bTreeBook1 106600

skipListBook1 35600

skipListJava1 50800

binarySearchTree2 32400

redBlackTreeBook2 45400

redBlackTreeJava2 41400

bTreeBook2 55300

skipListBook2 38300

skipListJava2 48700

binarySearchTree4 37600

redBlackTreeBook4 101700

redBlackTreeJava4 70300

bTreeBook4 78100

skipListBook4 66800

skipListJava4 101800

binarySearchTree8 45200

redBlackTreeBook8 66400

redBlackTreeJava8 70700

bTreeBook8 83600

skipListBook8 280200

skipListJava8 216600

10 element deletion times in nanoseconds

binarySearchTree1 118900

redBlackTreeBook1 133600

redBlackTreeJava1 149000

skipListBook1 129200

skipListJava1 296600

binarySearchTree2 170500

redBlackTreeBook2 325900

redBlackTreeJava2 176500

skipListBook2 110100

skipListJava2 126700

binarySearchTree4 85900

redBlackTreeBook4 100300

redBlackTreeJava4 142500

skipListBook4 151100

skipListJava4 242100

binarySearchTree8 85500

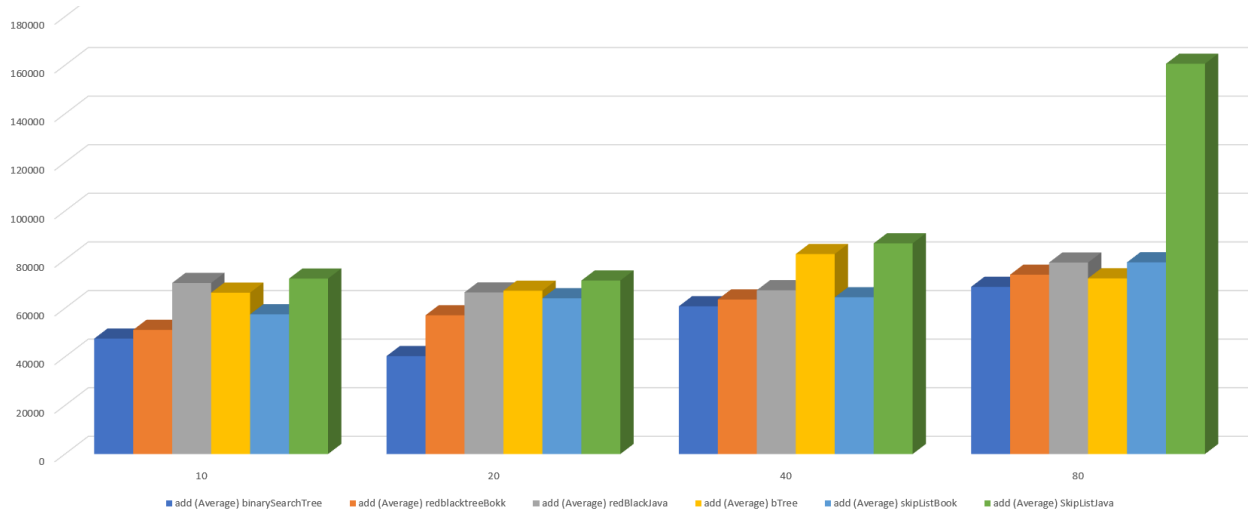
redBlackTreeBook8 83200

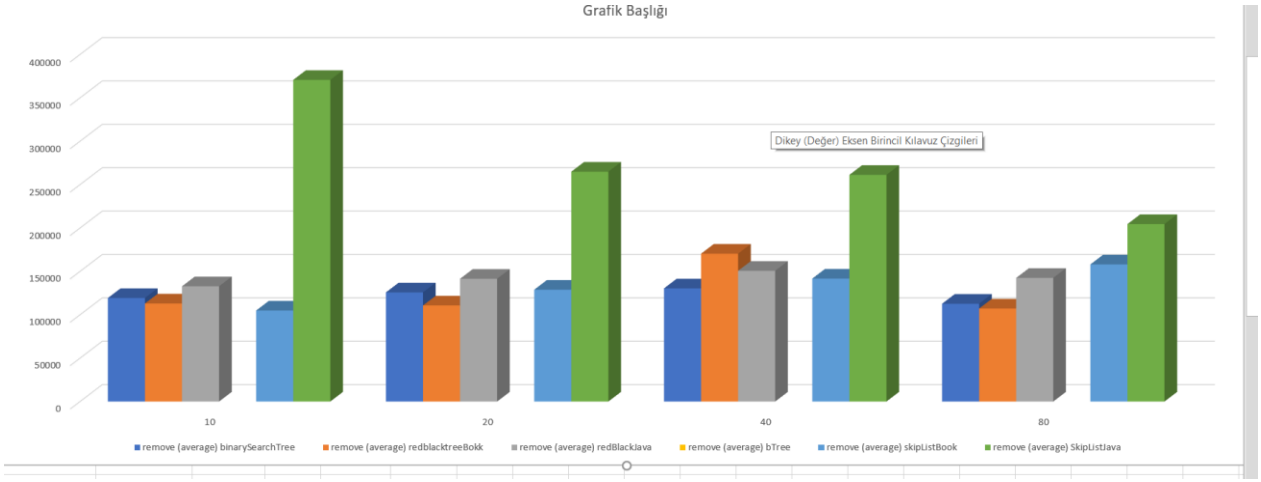
redBlackTreeJava8 239500

skipListBook8 145400

skipListJava8 240900

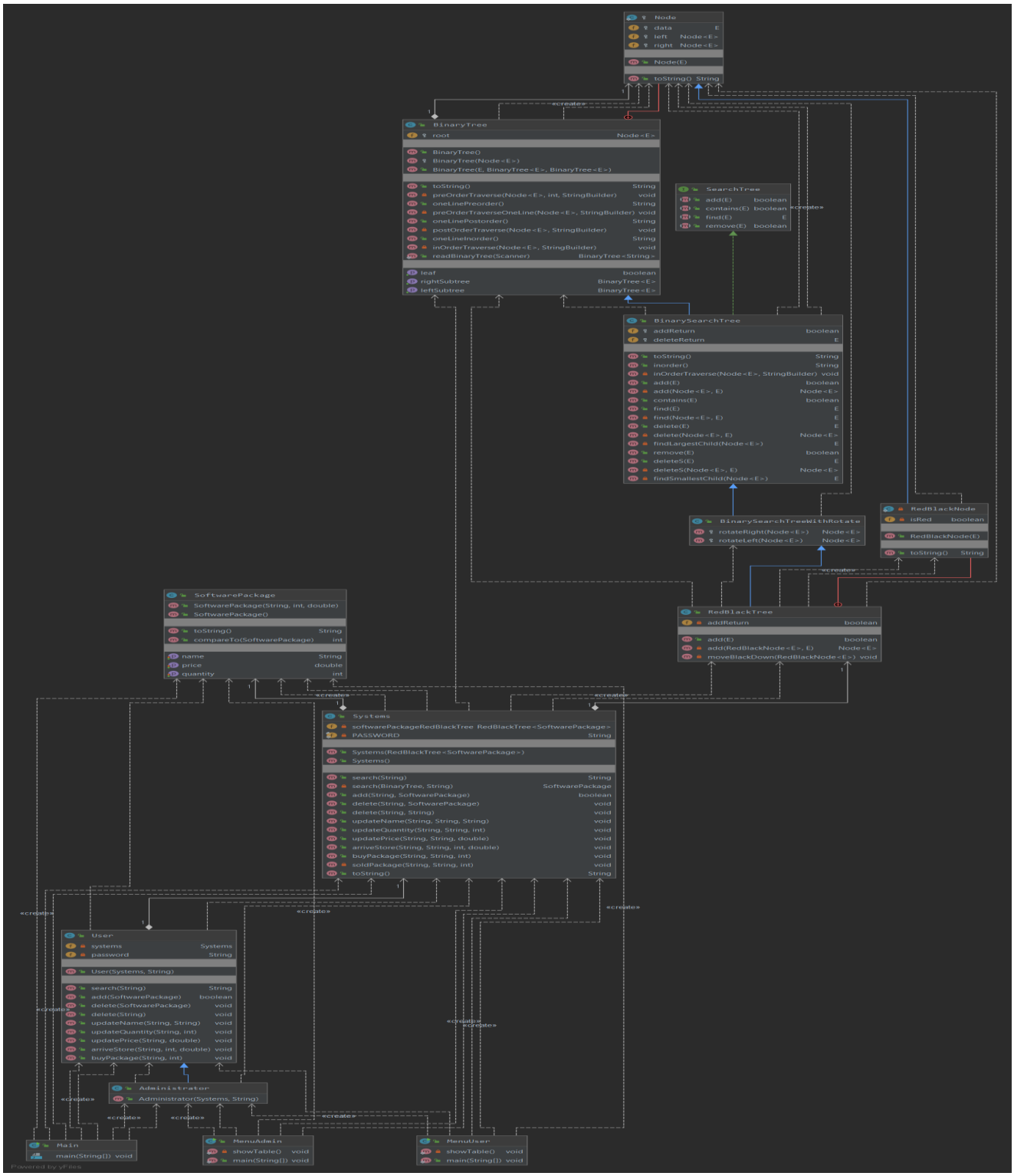
	add (Average)					
	binarySearchTree	redblacktreeBokk	redBlackJava	bTree	skipListBook	SkipListJava
10	47500	51100	70400	66400	57500	72200
20	40300	57100	66500	67200	64100	71400
40	60800	63600	67400	82300	64500	86700
80	68800	73800	78800	72300	78900	160600
	remove (average)					
	binarySearchTree	redblacktreeBokk	redBlackJava	bTree	skipListBook	SkipListJava
10	119300	113000	132800		104900	370700
20	125800	110700	141500		128900	264900
40	130400	170400	150700		141700	261200
80	112700	107200	142500		158000	204500

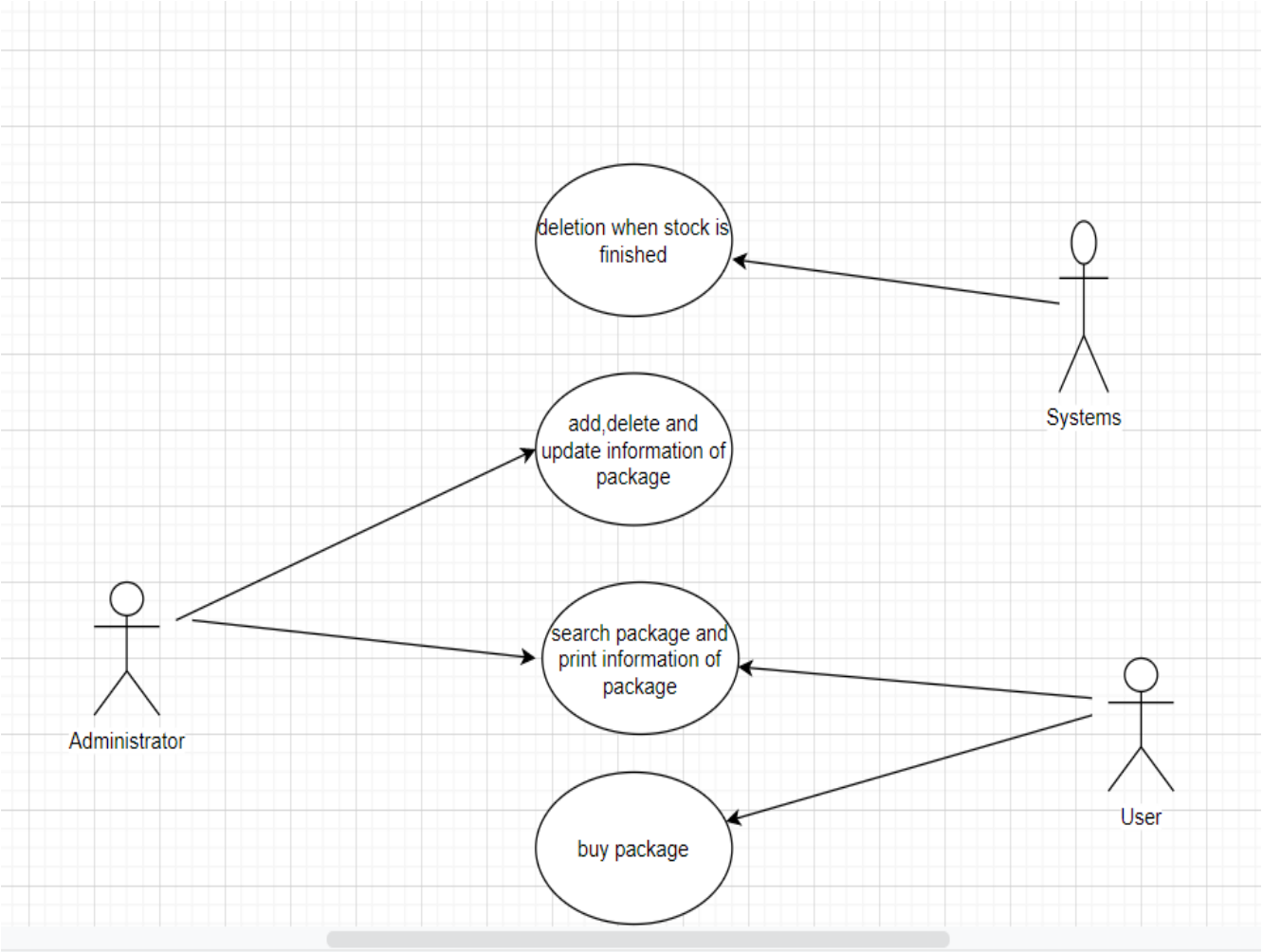



Grafik Başlığı
400000
350000
300000
250000
200000
150000
100000
50000
0
Dikey (Değer) Eksen Birincil Kılavuz Çizgileri
10
20
40
80
remove (average) binarySearchTree
remove (average) redblacktreeBokk
remove (average) redBlackJava
remove (average) bTree
remove (average) skipListBook
remove (average) SkipListJava

Part4

Class Diagram





Problem Solution Approach

I made a package class so that my red black tree in the system class can easily handle the packages.

I made it easy by making my tree a package class.

I determined which one can use which methods in the user and administrator classes with a password. I inherited all methods in the user class and inherited it to the admin class.

When the package is out of stock, I have enabled the system to automatically delete this package information from the tree.

Test Case and Running Comand and Result

I have shown below how all methods in the system class work in the admin and user menu.

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.3\lib\idea_rt.jar=52828:C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.3\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\foguz\IdeaProjects\HW7PART4\out\production\HW7PART4 MenuAdmin

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

s

Enter software name

Narton

there is not this package please try again

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

s

Enter software name

Narton 4.5

SoftwarePackage{name='Narton 4.5', quantity=200, price=190.78}

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

a

Enter software name

intelij

Enter qunatity

78

Enter software price

9,5

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name=' Narton 5.5', quantity=200, price=2000.78}

Red: SoftwarePackage{name='Narton 4.5', quantity=200, price=190.78}

-

-

Red: SoftwarePackage{name='intelij', quantity=78, price=9.5}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

d

Enter software name

Narton 4.5

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name='Narton 5.5', quantity=200, price=2000.78}

-

Red: SoftwarePackage{name='intelij', quantity=78, price=9.5}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

un

Enter software old name

Narton 5.5

Enter software new name

netbeans

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name='netbeans', quantity=200, price=2000.78}

-

Red: SoftwarePackage{name='intelij', quantity=78, price=9.5}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

uq

Enter software name

netbeans

Enter qunatity

1

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name='netbeans', quantity=1, price=2000.78}

-

Red: SoftwarePackage{name='intelij', quantity=78, price=9.5}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

up

Enter software name

netbeans

Enter software price

5

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name='netbeans', quantity=1, price=5.0}

-

Red: SoftwarePackage{name='intelij', quantity=78, price=9.5}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

as

Enter software name

netbeans

Enter qunatity

78

Enter software price

5

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name='netbeans', quantity=79, price=5.0}

-

Red: SoftwarePackage{name='intelij', quantity=78, price=9.5}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

b

Enter software name

netbeans

Enter qunatity

4

administrator can not buy package

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

q

Process finished with exit code 0

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.3\lib\idea_rt.jar=52700:C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.3\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\foguz\IdeaProjects\HW7PART4\out\production\HW7PART4 MenuUser

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name=' Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name=' Narton 4.5', quantity=200, price=190.78}

-

Red: SoftwarePackage{name=' Narton 5.5', quantity=200, price=2000.78}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

s

Enter software name

Narton 4.5

SoftwarePackage{name='Narton 4.5', quantity=200, price=190.78}

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

a

Enter software name

intelij

Enter qunatity

78

Enter software price

6

user does not add

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

D

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

d

Enter software name

Narton 4.5

Users do not remove

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

un

Enter software old name

Narton 4.5

Enter software new name

intelij

User do not update Name,price,quatity

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

uq

Enter software name

Narton 4.5

Enter qunatity

78

User do not update Name,price,quatity

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

up

Enter software name

Narton 4.5

Enter software price

78

User do not update Name,price,quatity

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

as

Enter software name

Narton 4.5

Enter qunatity

45

Enter software price

89

user do not this action

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

b

Enter software name

Narton 4.5

Enter qunatity

78

78 Narton 4.5 buys

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name='Narton 4.5', quantity=122, price=190.78}

-

Red: SoftwarePackage{name=' Narton 5.5', quantity=200, price=2000.78}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

b

Enter software name

Narton 4.5

Enter qunatity

100

100 Narton 4.5 buys

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Black: SoftwarePackage{name='Narton 4.5', quantity=22, price=190.78}

-

Red: SoftwarePackage{name=' Narton 5.5', quantity=200, price=2000.78}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

b

Enter software name

Narton 4.5

Enter qunatity

22

22 Narton 4.5 buys

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

?

Black: SoftwarePackage{name='Adobe Photoshop 6.2', quantity=200, price=19650.78}

Black: SoftwarePackage{name=' Adobe Flash 4.0', quantity=2900, price=1960.78}

Red: SoftwarePackage{name=' Adobe Flash 3.3', quantity=2000, price=18460.78}

-

-

Red: SoftwarePackage{name=' Adobe Photoshop 6.0', quantity=2000, price=1450.78}

-

-

Red: SoftwarePackage{name=' Narton 5.5', quantity=200, price=2000.78}

-

-

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

b

Enter software name

Narton 5

Enter qunatity

78

out of stock

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****

****

?: Display

q: Quit

s: Search

a: Add

d: Delete

un: Update Name

uq: Update Quantity

up: Update Price

as: Arrive Store

b: Buy Package

****