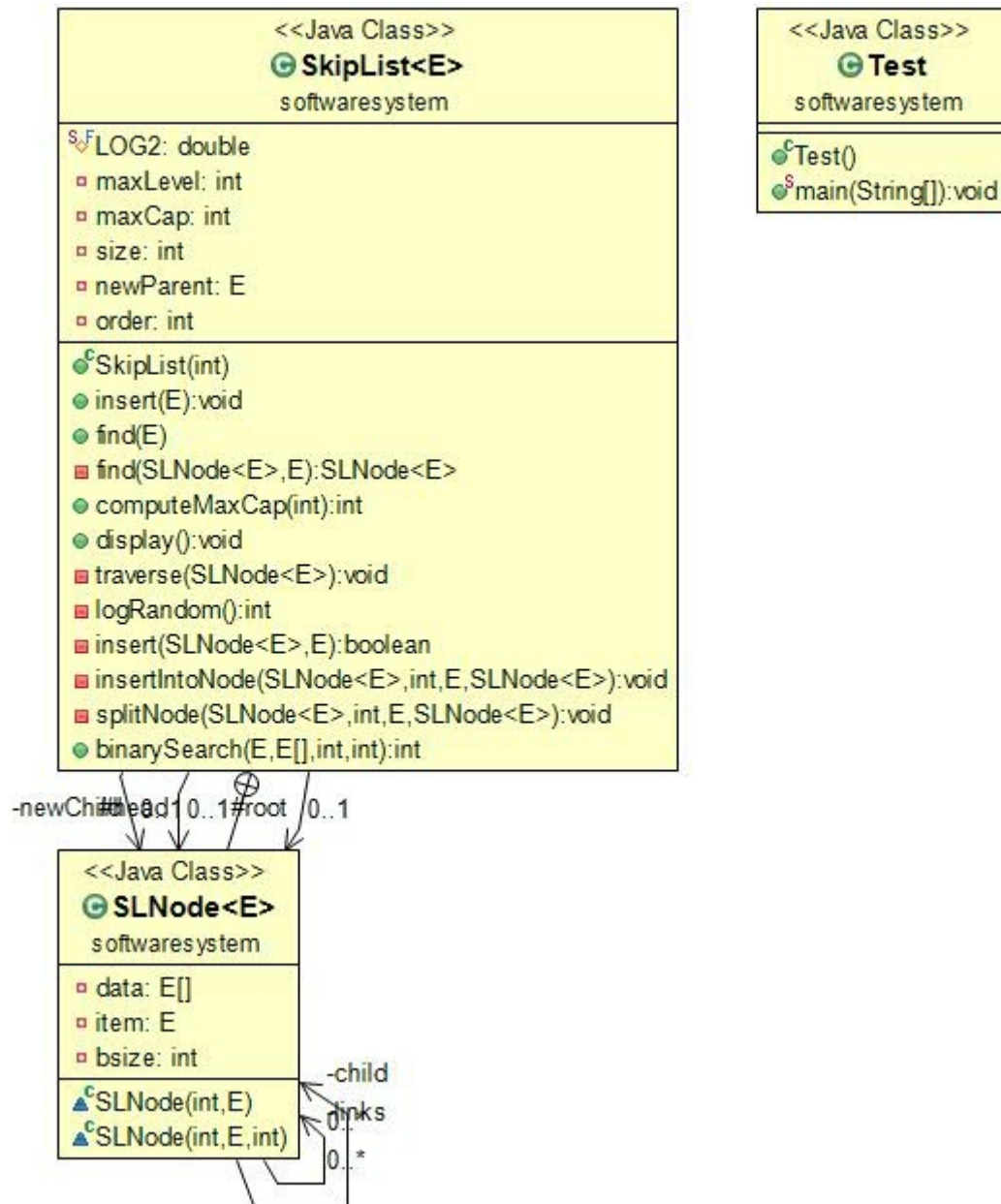


GTU Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 7 Report

Mert Can BEŞİRLİ
1801042663

#Question2#

1. CLASS DIAGRAMS



2. PROBLEM SOLUTION APPROACH

In this problem, modify skip list implementation so that each node in the lowest-level list keeps elements instead of just one element as in B-tree node. In the addition process, if the root is null, the normal node is added and the first node of tree B is created. In the other case, the level is determined first and then the B tree is formed according to the level in the links formed up to the minimum level of the head. For example, a B tree is formed at the link zero of the head and a B tree is formed at the link one. In this way, it continues up to the maximum level of the head.

3. TEST CASES

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
T1	Check insert element and order is 4	Inserting 10 to 1 element	Head Links 0: 2, 5, 8 1 3, 4 6, 7 9, 10 Links 1: 2, 5, 7 Links 2: 2, 5 Links 3:	Creating head and links, links is B tree	As Excepted	Pass
T2	Check insert element and order is 5	Inserting 10 to 1 element	Head Links 0: 5, 8 1, 2, 3, 4 6, 7 9, 10 Links 1:	Creating head and links, links is B tree	As Excepted	Pass

			6			
			Links 2: Links 3:			

4. RUNNING AND RESULTS

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Test$ java Test
Head
Links 0:
2, 5, 8
1

3, 4

6, 7

9, 10

Links 1:
2, 5, 7

Links 2:
2, 5

Links 3:
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Test$ ^C
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Test$ java Test
Head
Links 0:
5, 8
1, 2, 3, 4

6, 7

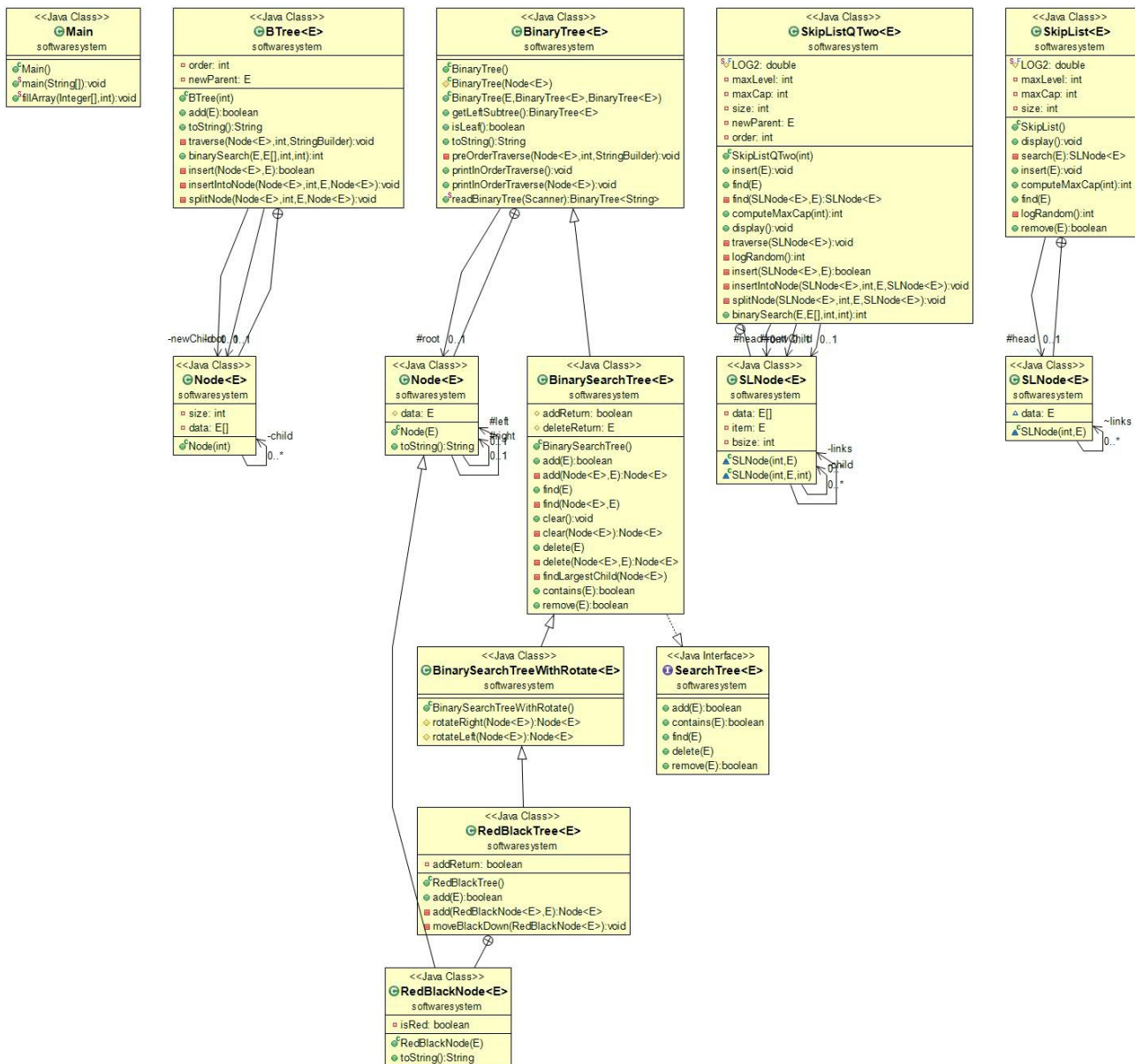
9, 10

Links 1:
6

Links 2:
Links 3:
```

#Question3#

1. CLASS DIAGRAMS



2. PROBLEM SOLUTION APPROACH

In this problem, compare performance of data structures, insert, extra 10 number insert, delete to four different problem sizes. Using data structures:

Regular Binary Search Tree

Red-Black Tree in book

Red-Black Tree in java

B-tree in book

Skip list in book

Skip list in java

Skip list in question 2

Measure running time and find average running time of this data structures.

3. TEST CASES

Test Case ID	Test Scenario	Test Steps	Test Data	Excepted Results	Actual Results	Pass/Fail
T1	Check size is 10000	Random number adding this data structures and 10 extra number, then delete all of them	Measure time	Showing measuring time of each data structures	As Excepted	Pass
T2	Check size is 20000	Random number adding this data structures and 10 extra number, then delete all of them	Measure time	Showing measuring time of each data structures	As Excepted	Pass
T3	Check size is 40000	Random number adding this data structures and 10 extra number, then delete all of them	Measure time	Showing measuring time of each data structures	As Excepted	Pass
T4	Check size	Random	Measure	Showing	As	Pass

	is 80000	number adding this data structures and 10 extra number, then delete all of them	time	measuring time of each data structures	Excepted	
--	----------	---	------	---	----------	--

4. RUNNING AND RESULTS

Note: Recompile with -Xlint:unchecked for details.

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

```
Time in milliseconds, binary search tree insert 10000 length: 39
```

```
Time in milliseconds, red black tree insert 10000 length: 50
```

```
Time in milliseconds, b tree insert 10000 length: 76
```

```
Time in milliseconds, skip list in book insert 10000 length: 130
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

```
Time in milliseconds, binary search tree insert 20000 length: 81
```

```
Time in milliseconds, red black tree insert 20000 length: 124
```

```
Time in milliseconds, b tree insert 20000 length: 159
```

```
Time in milliseconds, skip list in book insert 20000 length: 361
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

```
Time in milliseconds, binary search tree insert 40000 length: 162
```

```
Time in milliseconds, red black tree insert 40000 length: 225
```

```
Time in milliseconds, b tree insert 40000 length: 362
```

```
Time in milliseconds, skip list in book insert 40000 length: 729
```

Note: Recompile with -Xlint:unchecked for details.

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

```
Time in milliseconds, binary search tree insert 80000 length: 536
```

```
Time in milliseconds, red black tree insert 80000 length: 468
```

```
Time in milliseconds, b tree insert 80000 length: 765
```

```
Time in milliseconds, skip list in book insert 80000 length: 1340
```


Note: Recompile with `-Xint:checkcast` for details.

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

Time in milliseconds, binary search tree insert 10000 length: 43

Time in milliseconds, red black tree insert 10000 length: 105

Time in milliseconds, b tree insert 10000 length: 178

Time in milliseconds, skip list in book insert 10000 length: 239

Time in nanoseconds, binary search in book extra random number insert 10000 length: 19773

Time in nanoseconds, red black tree extra random number insert 10000 length: 31224

Time in nanoseconds, b tree extra random number insert 10000 length: 746749

Time in nanoseconds, skip list extra random number insert 10000 length: 39227

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

Time in milliseconds, binary search tree insert 20000 length: 83

Time in milliseconds, red black tree insert 20000 length: 154

Time in milliseconds, b tree insert 20000 length: 179

Time in milliseconds, skip list in book insert 20000 length: 475

Time in nanoseconds, binary search in book extra random number insert 20000 length: 73507

Time in nanoseconds, red black tree extra random number insert 20000 length: 89217

Time in nanoseconds, b tree extra random number insert 20000 length: 106175

Time in nanoseconds, skip list extra random number insert 20000 length: 129481

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

```
Time in milliseconds, binary search tree insert 10000 length: 42
Time in milliseconds, red black tree insert 10000 length: 131
Time in milliseconds, b tree insert 10000 length: 389
Time in milliseconds, skip list in book insert 10000 length: 340
Time in milliseconds, skip list in question two insert 10000 length: 363
Time in milliseconds, skip list in java insert 10000 length: 339
Time in milliseconds, red black tree in java insert 10000 length: 139
Time in nanoseconds, binary search in book extra random number insert 10000 length: 23683
Time in nanoseconds, red black tree extra random number insert 10000 length: 30528
Time in nanoseconds, b tree extra random number insert 10000 length: 63210
Time in nanoseconds, skip list extra random number insert 10000 length: 170804
Time in nanoseconds, skip list in question two extra random number insert 10000 length: 190905
Time in nanoseconds, skip list in java extra random number insert 10000 length: 55230
Time in nanoseconds, red black tree in java extra random number insert 10000 length: 23830
Time in milliseconds, binary search tree delete 10000 length: 35
Time in milliseconds, skip list delete 10000 length: 397
Time in milliseconds, skip list in java delete 10000 length: 67
Time in milliseconds, red black tree in java delete 10000 length: 0
```

```
ubuntu@ubuntu-VirtualBox:~/Masaüstü/Tree/Q3$ java Main
```

```
Time in milliseconds, binary search tree insert 20000 length: 82
Time in milliseconds, red black tree insert 20000 length: 208
Time in milliseconds, b tree insert 20000 length: 207
Time in milliseconds, skip list in book insert 20000 length: 356
Time in milliseconds, skip list in question two insert 20000 length: 759
Time in milliseconds, skip list in java insert 20000 length: 275
Time in milliseconds, red black tree in java insert 20000 length: 111
Time in nanoseconds, binary search in book extra random number insert 20000 length: 89894
Time in nanoseconds, red black tree extra random number insert 20000 length: 119437
Time in nanoseconds, b tree extra random number insert 20000 length: 131690
Time in nanoseconds, skip list extra random number insert 20000 length: 401213
Time in nanoseconds, skip list in question two extra random number insert 20000 length: 338922
Time in nanoseconds, skip list in java extra random number insert 20000 length: 176408
Time in nanoseconds, red black tree in java extra random number insert 20000 length: 98792
Time in milliseconds, binary search tree delete 20000 length: 2
Time in milliseconds, skip list delete 20000 length: 899
Time in milliseconds, skip list in java delete 20000 length: 35
Time in milliseconds, red black tree in java delete 20000 length: 0
```

```
ubuntu@ubuntu-virtualbox: ~/nausatu/tree/q35$ java Main
```

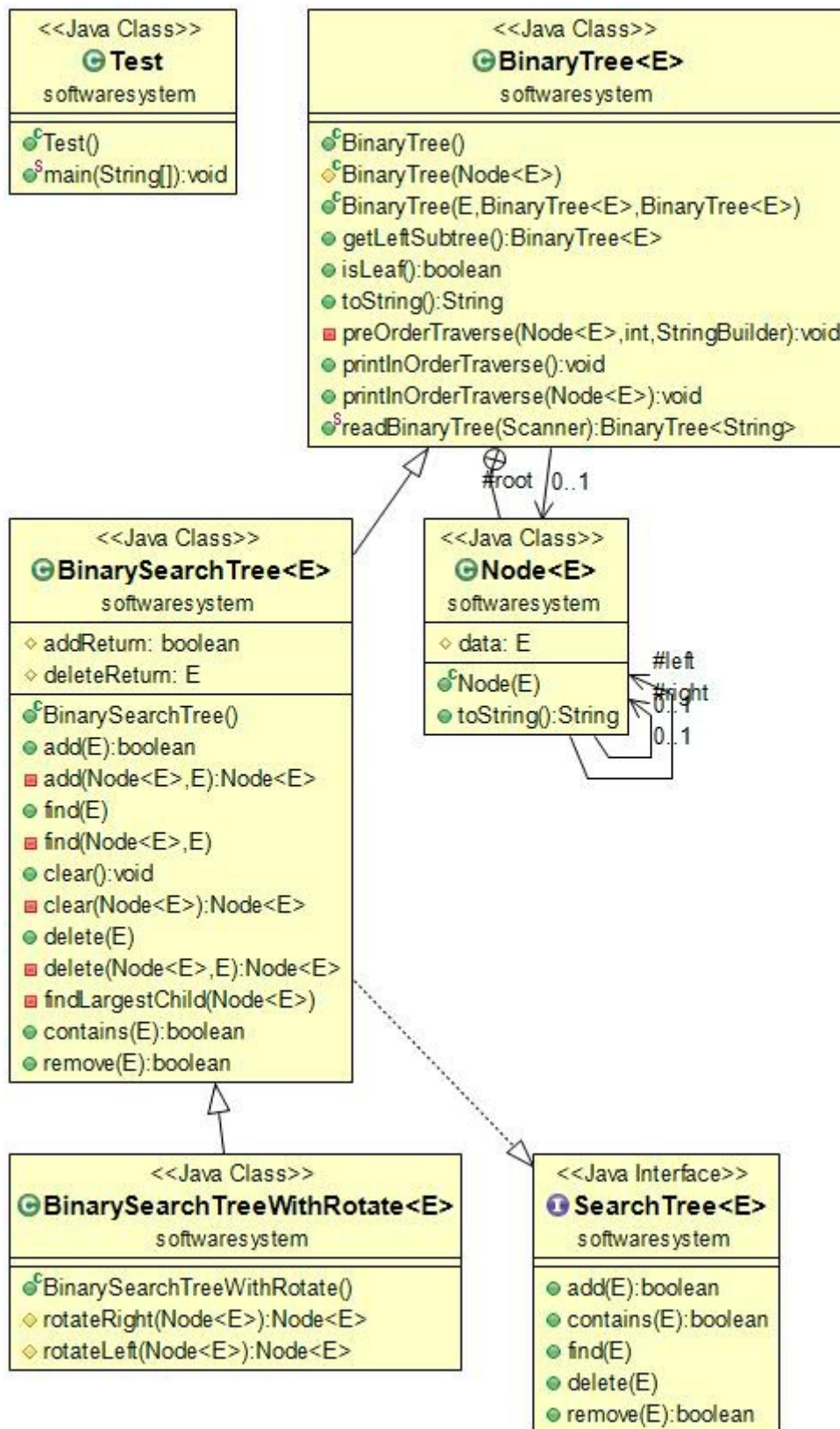
```
Time in milliseconds, binary search tree insert 40000 length: 173
Time in milliseconds, red black tree insert 40000 length: 332
Time in milliseconds, b tree insert 40000 length: 454
Time in milliseconds, skip list in book insert 40000 length: 1154
Time in milliseconds, skip list in question two insert 40000 length: 1575
Time in milliseconds, skip list in java insert 40000 length: 523
Time in milliseconds, red black tree in java insert 40000 length: 239
Time in nanoseconds, binary search in book extra random number insert 40000 length: 92445
Time in nanoseconds, red black tree extra random number insert 40000 length: 102072
Time in nanoseconds, b tree extra random number insert 40000 length: 148193
Time in nanoseconds, skip list extra random number insert 40000 length: 172842
Time in nanoseconds, skip list in question two extra random number insert 40000 length: 251191
Time in nanoseconds, skip list in java extra random number insert 40000 length: 185569
Time in nanoseconds, red black tree in java extra random number insert 40000 length: 108040
Time in milliseconds, binary search tree delete 40000 length: 7
Time in milliseconds, skip list delete 40000 length: 2775
Time in milliseconds, skip list in java delete 40000 length: 43
Time in milliseconds, red black tree in java delete 40000 length: 0
```

```
ubuntu@ubuntu-virtualbox: ~/nausatu/tree/q36$ java Main
```

```
Time in milliseconds, binary search tree insert 80000 length: 565
Time in milliseconds, red black tree insert 80000 length: 610
Time in milliseconds, b tree insert 80000 length: 920
Time in milliseconds, skip list in book insert 80000 length: 1897
Time in milliseconds, skip list in question two insert 80000 length: 2565
Time in milliseconds, skip list in java insert 80000 length: 844
Time in milliseconds, red black tree in java insert 80000 length: 559
Time in nanoseconds, binary search in book extra random number insert 80000 length: 34896
Time in nanoseconds, red black tree extra random number insert 80000 length: 41909
Time in nanoseconds, b tree extra random number insert 80000 length: 58023
Time in nanoseconds, skip list extra random number insert 80000 length: 147970
Time in nanoseconds, skip list in question two extra random number insert 80000 length: 270305
Time in nanoseconds, skip list in java extra random number insert 80000 length: 47304
Time in nanoseconds, red black tree in java extra random number insert 80000 length: 23001
Time in milliseconds, binary search tree delete 80000 length: 14
Time in milliseconds, skip list delete 80000 length: 5068
Time in milliseconds, skip list in java delete 80000 length: 59
Time in milliseconds, red black tree in java delete 80000 length: 0
```

#Question4#

1. CLASS DIAGRAMS



2. PROBLEM SOLUTION APPROACH

In this problem, menu driven program for managing a software store. There are three admins and three users, I have set them. The user needs to enter the password correctly in order to add, remove and update the system (password is cse222). Searching can be done in two ways, by name and amount, these are in the menu options. I used the map structure to understand what software package quantity and price. Within the set structure for admin and users. There are 3 trees with name, quantity and price, and they keep the elements here. Trees are balanced trees, rotating takes place in case of any imbalance. I made the update part of my name on the price. Menu options:

- Adding process
- Removing process
- Updating process
- Search by name
- Search by quantity
- Users buy process
- Quit menu

3. TEST CASES

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
T1	Check admin password correctly then adding process	Menu choose A option	Enter of adding software name: Oracle Enter of quantity for adding software name: 9 Enter of price for adding software name: 45	Enter software package, successfully then show all element the trees	As Excepted	Pass
T2	Check	Menu	B	Removing	As	Pass

	removing process	choose B option	Enter of removing software name: Oracle	choosing package, succesfully	Excepted	
T3	Check updating process	Menu choose C option	C Enter of update software element's name: Norton 4.5 Enter of software element's updating price: 75	Choosing update package's price is changed, old value remove in tree and adding new price, successfull y,then showing all tree elements	As Excepted	Pass
T4	Check Search By name	Choose D option	C Enter of update software element's name: Norton 4.5 Enter of software element's updating price: 75	Choose which user is process and searching package name and showing result	As Excepted	Pass
T5	Check Search by quanti y	Choose E option	E Which user wants to take process? [User2, User1, User3] User1 Enter of	Choose which user is process and searching by quantity and showing result	As Excepted	Pass

			software quantity for search: 3			
T6	Check users buy process and update buying software packages	Choose F option	Choose which user is process and searching package name and showing result	Choose which user is process and if quantity is bigger than 1, quantity is decrease 1, otherwise package is removed	As Excepted	Pass
T7	Quit	Check Q option	Quit	Ending program	As Excepted	Pass

4. RUNNING AND RESULTS

```

Software name:
Adobe Flash 3.3 Adobe Flash 4.0 Adobe Photoshop 6.0 Adobe Photoshop 6.2 Norton 4.5 Norton 5.5
Software price:
5 10 15 25 35 65
Software quantity:
1 2 3 4 5 6
Which administrator wants to take action?
[Admin3, Admin2, Admin1]
Admin3
Enter password for enter software system(password is 'cse222'):
cse222
A->Adding process by admin
B->Removing process by admin
C->Update process by admin
D->Search by name from user
E->Search by quantity
F->Users buy process
Q->Quit
A
Enter of adding software name:
Oracle
Enter of quantity for adding software name:
9
Enter of price for adding software name:
45
Software name:
Adobe Flash 3.3 Adobe Flash 4.0 Adobe Photoshop 6.0 Adobe Photoshop 6.2 Norton 4.5 Norton 5.5 Oracle
Software price:
5 10 15 25 35 45 65
Software quantity:
1 2 3 4 5 6 9
A->Adding process by admin

```

```
A->Adding process by admin
B->Removing process by admin
C->Update process by admin
D->Search by name from user
E->Search by quantity
F->Users buy process
Q->Quit
B
Enter of removing software name:
Oracle
Software name:
Adobe Flash 3.3 Adobe Flash 4.0 Adobe Photoshop 6.0 Adobe Photoshop 6.2 Norton 4.5 Norton 5.5
Software price:
5 10 15 25 35 65
Software quantity:
1 2 3 4 5 6
```

```
A->Adding process by admin
B->Removing process by admin
C->Update process by admin
D->Search by name from user
E->Search by quantity
F->Users buy process
Q->Quit
C
Enter of update software element's name:
Norton 4.5
Enter of software element's updating price:
75
Software name:
Adobe Flash 3.3 Adobe Flash 4.0 Adobe Photoshop 6.0 Adobe Photoshop 6.2 Norton 4.5 Norton 5.5
Software price:
10 15 25 35 65 75
Software quantity:
1 2 3 4 5 6
```

```
A->Adding process by admin
B->Removing process by admin
C->Update process by admin
D->Search by name from user
E->Search by quantity
F->Users buy process
Q->Quit
D
Which user wants to take process?
[User2, User1, User3]
User1
Enter of software name for search:
Norton 5.5
Software name : Norton 5.5, quantity : 2, price : 25A->Adding process by admin
```

```
A->Adding process by admin
B->Removing process by admin
C->Update process by admin
D->Search by name from user
E->Search by quantity
F->Users buy process
Q->Quit
E
Which user wants to take process?
[User2, User1, User3]
User1
Enter of software quantity for search:
3
Software name: Adobe Flash 3.3, quantity : 3, price: 65
```



```
A->Adding process by admin
B->Removing process by admin
C->Update process by admin
D->Search by name from user
E->Search by quantity
F->Users buy process
Q->Quit
F
Which user wants to take process?
[User2, User1, User3]
User1
Software name:
Adobe Flash 3.3 Adobe Flash 4.0 Adobe Photoshop 6.0 Adobe Photoshop 6.2 Norton 4.5 Norton 5.5
Enter of buying software name:
Norton 4.5
Software name:
Adobe Flash 3.3 Adobe Flash 4.0 Adobe Photoshop 6.0 Adobe Photoshop 6.2 Norton 4.5 Norton 5.5
Software price:
5 10 15 25 35 65
Software quantity:
1 2 3 5 6
A->Adding process by admin
```