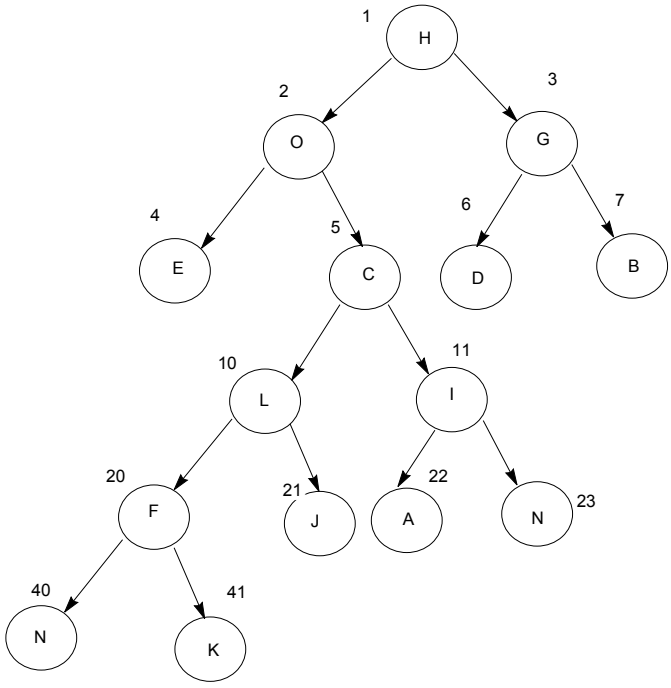
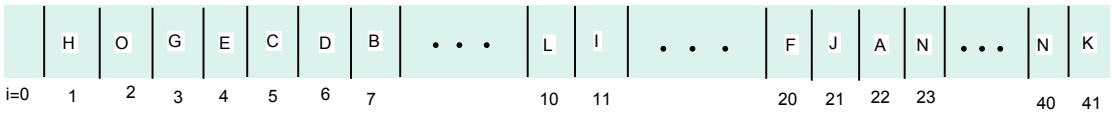
Polina Anis'kina
ID: 26991092
COMP-352-F
November 27-2016
Assignment #3-4 (Theory Part)

Question 1

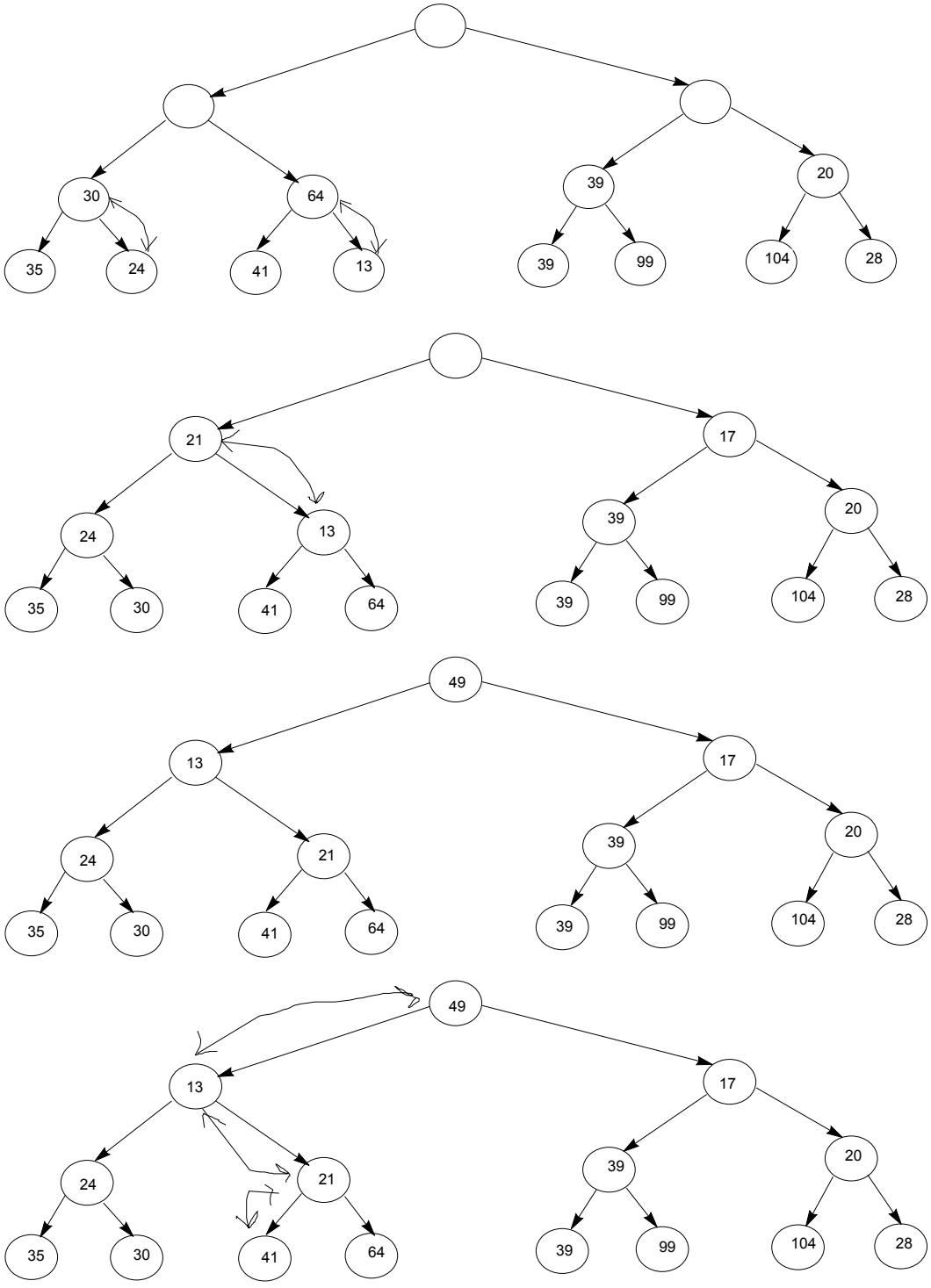


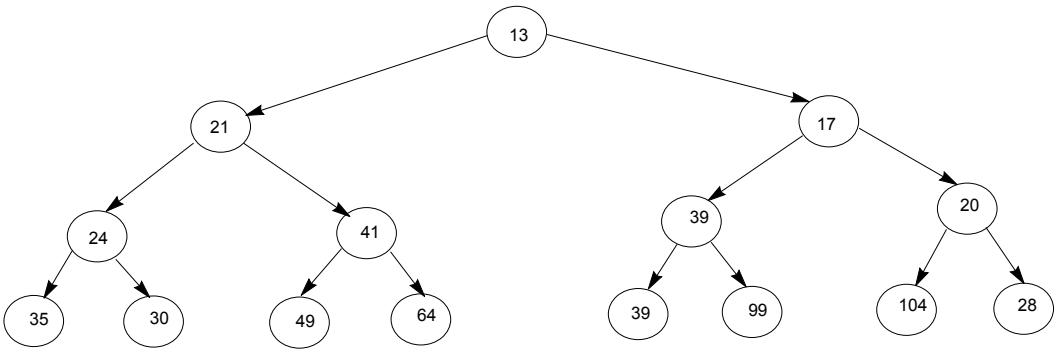
Question 2



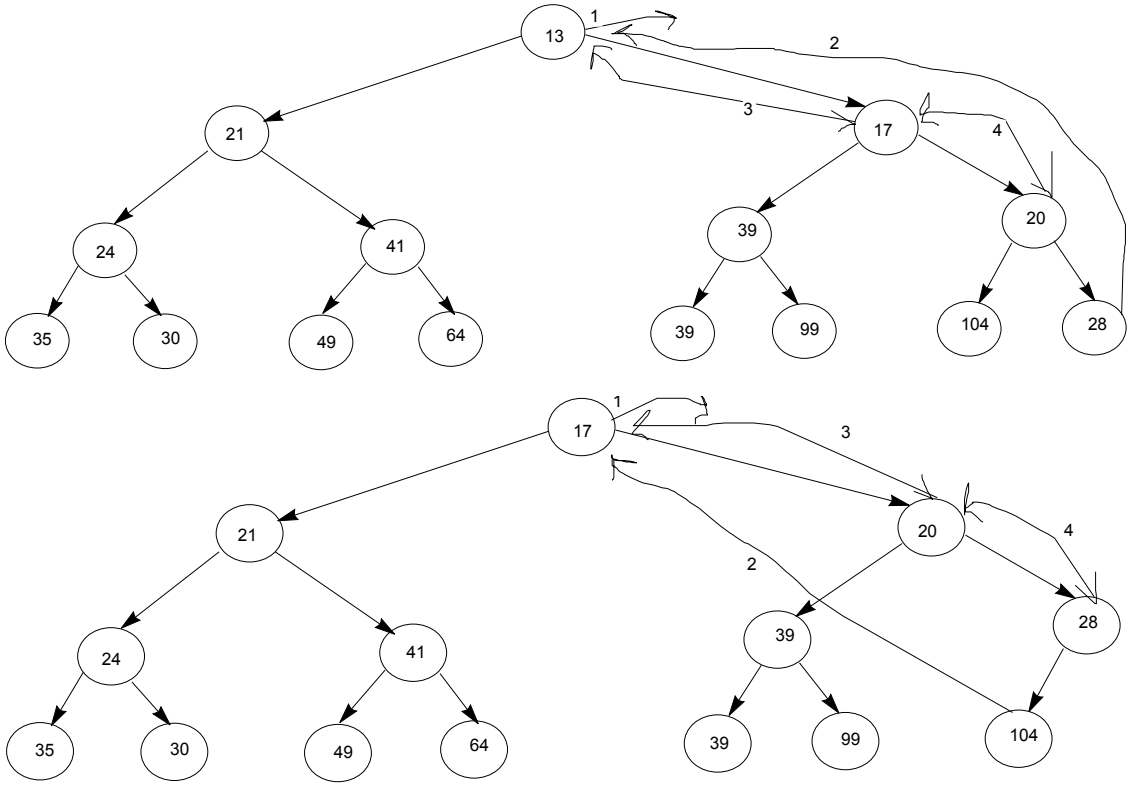
Question 3

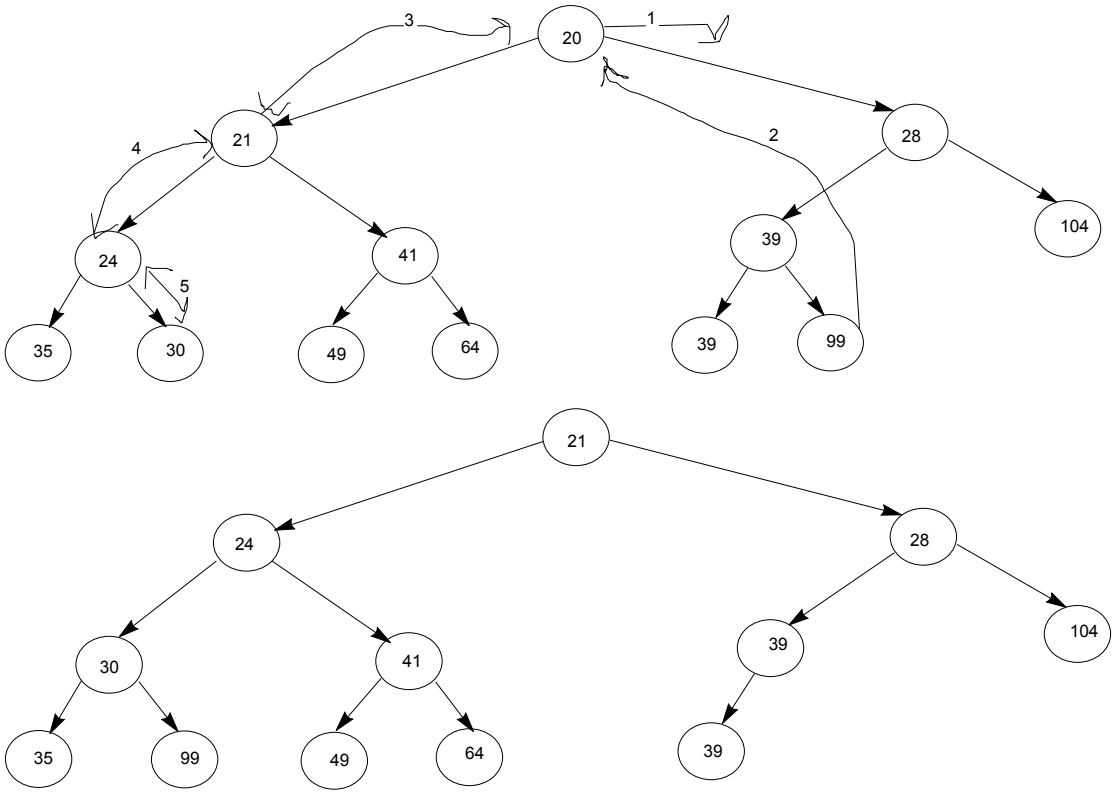
```
The list : [35, 24, 41, 13, 39, 99, 104, 28, 30, 64, 39, 20, 21, 17, 49] bottom up
```


30
64
39
20
35
24
41
13
39
99
104
28
21
17
24
13
39
20
35
30
41
64
39
99
104
28
49
13
17
24
21
39
20
35
30
41
64
39
99
104
28
49
13
17
24
21
39
20
35
30
41
64
39
99
104
28

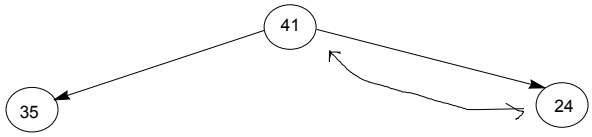


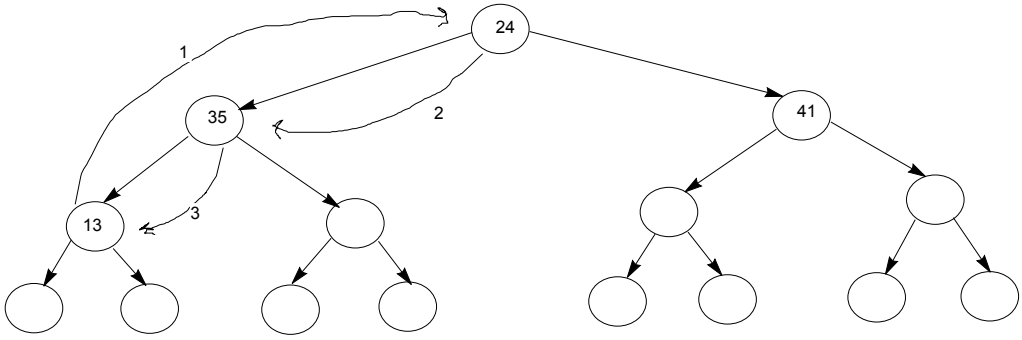
now we removeMin () 3 times :

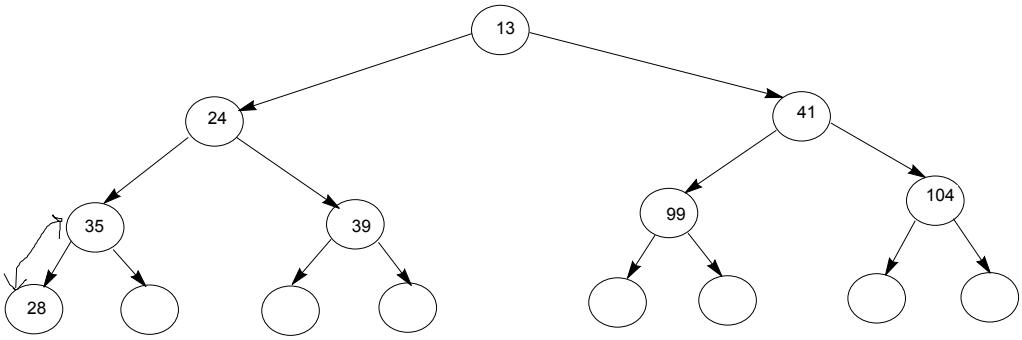


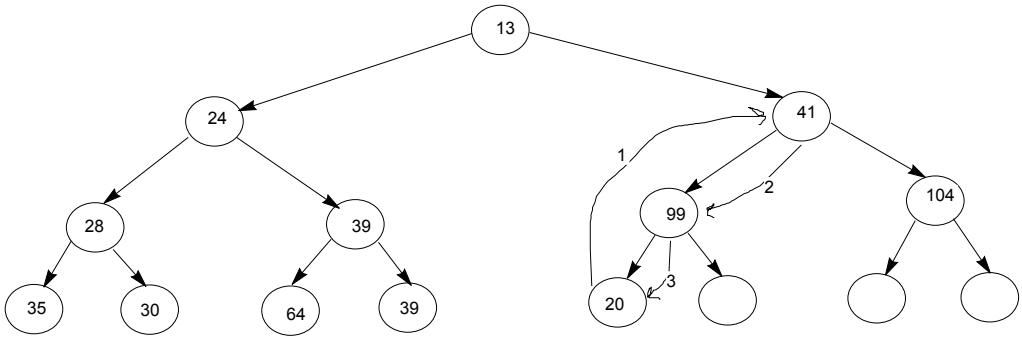

3
1
20
21
28
4
2
24
41
39
104
5
35
30
49
64
39
99
21
24
28
30
41
39
104
35
99
49
64
39

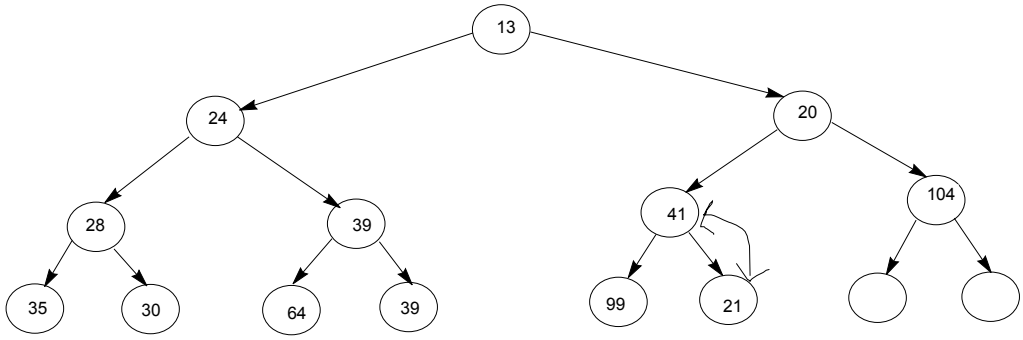
Question 4

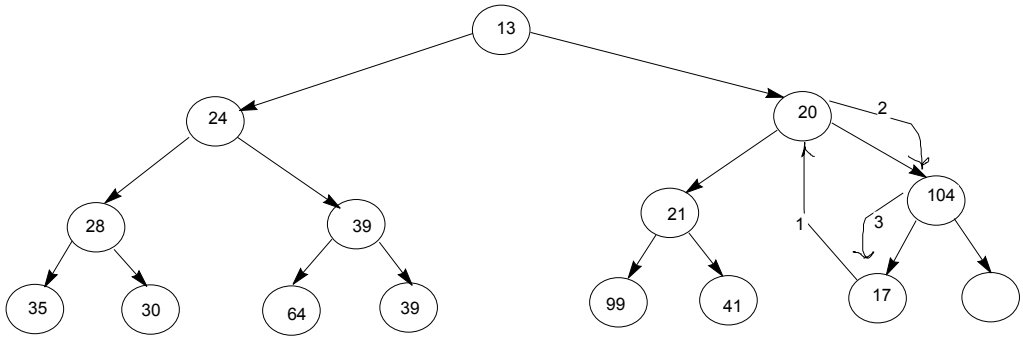

41
35
24

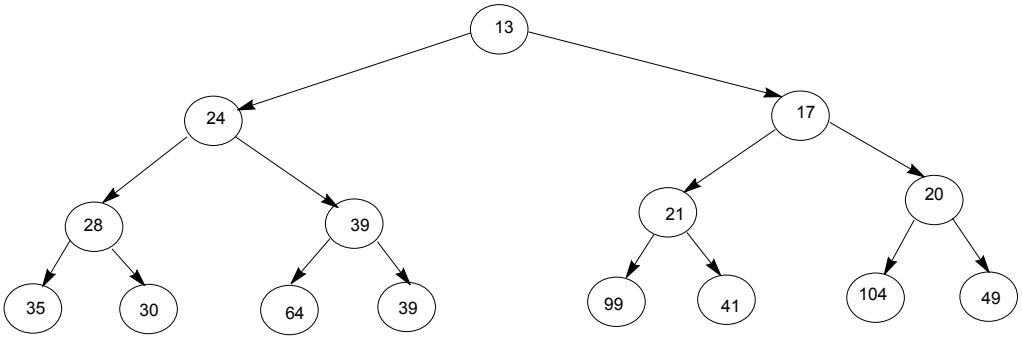

24
1
35
2
41
13
3


13
24
41
35
39
99
104
28


13
24
41
1
2
28
39
99
104
3
35
30
64
39
20


13
24
20
28
39
41
104
35
30
64
39
99
21


13
24
20
2
28
39
21
104
1
3
35
30
64
39
99
41
17



Question 5

```
Algorithm DepthNodes( T, v )
Input: tree T and v is a node of T
Output: the depth of all the nodes of a tree
-----------------------------------------------
if T.isRoot(v) then
    setDepth(v,o)
else
    setDepth(v, 1+getDepth(T.parent(v)))

children ⟵ T.children(v)

while children.hasNext() do
    child ⟵ children.next()
    DepthNodes(T, child)
```

a) The time complexity of this algorithms is O(n) because in the if and else statement is O(1), and the while loop will be performing n times which is equal to the total number of the tree.

b) The best possible complexity is O(n), because at each node it does a constant number of operations and the number of nodes is n, so it is the best time complexity that possible to achieve.

c) This algorithm achieve this best complexity.

Question 6

i) This proposal would have an advantage because it is going to give the removed key right away, and we dont have to search the whole hash table for the key.

ii) This proposal would be disadvantegeous because in order to restore the original insert it will have to shift the entries up by 1 index. Due to this search algorithm will be improved but the removed algorithm would be lessened.

Question 7

```
Mod[32, 13]
```

```
6
```

```
Mod[147, 13]
```

4

```
Mod[265, 13]
```

5

```
Mod[195, 13]
```

0

```
Mod[207, 13]
```

12

```
Mod[180, 13]
```

11

```
Mod[21, 13]
```

8

```
Mod[16, 13]
```

3

```
Mod[189, 13]
```

7

```
Mod[202, 13]
```

7

```
Mod[91, 13]
```

0

```
Mod[94, 13]
```

3

```
Mod[162, 13]
```

6

```
Mod[75, 13]
```

10

```
Mod[37, 13]
```

11

```
Mod[77, 13]
```

12

```
Mod[81, 13]
```

```
3
```

```
Mod[48, 13]
```

```
9
```

```
 i = 0   91  195
 i = 1
 i = 2
 i = 3   81   94  16
 i = 4  147
 i = 5  265
 i = 6  162  32
 i = 7  202  189
 i = 8   21
 i = 9   48
i = 10  75  147
i = 11  37  180
i = 12  77  207
```

The maximum number of collisions is 3.

Question 8

Such proposal is senseless because not prime number will cause more collisions. Only prime number of the size of the array will recude the load factor and hence the number of collisions.

Question 9

```
Mod[38, 19]
```

```
0
```

```
Mod[15, 19]
```

```
15
```

```
Mod[43, 19]
```

```
5
```

```
Mod[22, 19]
```

```
3
```

```
Mod[71, 19]
```

```
14
```

```
Mod[8, 19]
```

```
8
```

```
Mod[28, 19]
```

```
9
```

```
Mod[37, 19]
```

```
18
```

```
Mod[19, 19]
```

```
0
```

i)

d (k) = 11 − k mod 11

d (19) = 11 − 19 mod 11 = 3

i = 0	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8	i = 9	i = 10	i = 11
38	□	□	22	□	43	19	□	8	28	□	□
collision (19)	□	□	collision (19)	□	□	□	□	□	□	□	□

ii) The largest cluster is 2
iii) The number of occured collisions is 2
iv) a=n/N=9/19=0.47

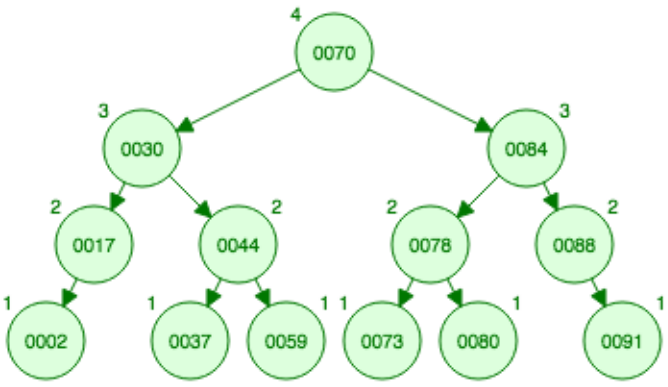
Question 10

i = 0	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8	i = 9	i = 10	i = 11	i = 1
95	19	□	□	□	□	□	□	□	□	29	30	12
collision (19)	□	□	□	□	□	□	□	□	□	AVAILABLE	□	□
□	□	□	□	□	□	□	□	□	□	□	□	□

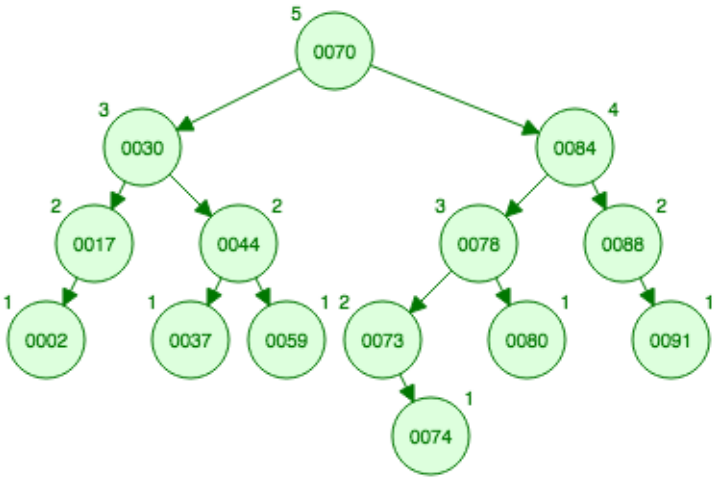
ii) The size of the longest cluster is 3. Remove is O(1) and put is O(n).
iii) 1 collision

Question 11

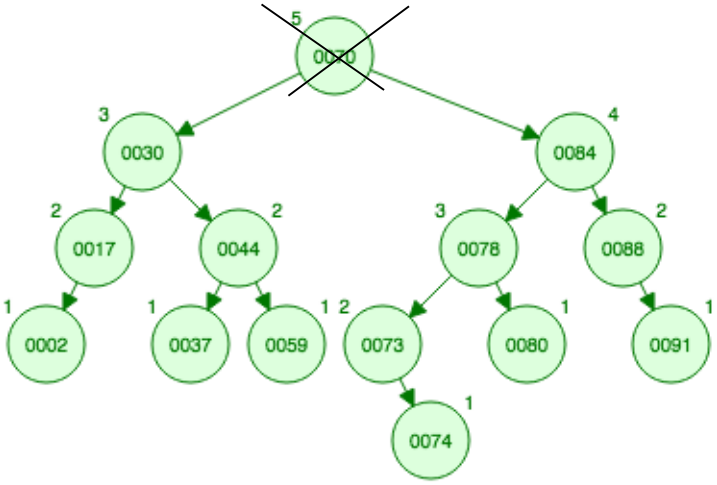
i) There are errors with the tree, the right side with value 84 has height of 6, and left side with value 30 has height of 4, which makes the difference = 2, and not 1. Also the value 2, is not supposed to be on the right side of 59, but on the left.
Here there is a right AVL tree:

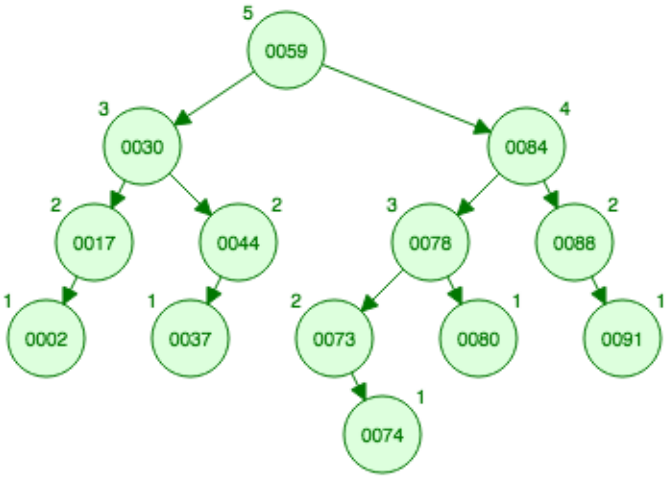


ii) AVL tree after put(74) is performed with complexity of O(log n):

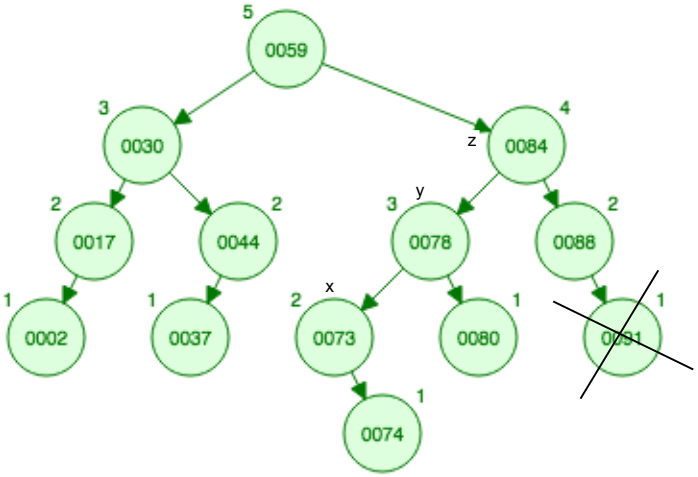


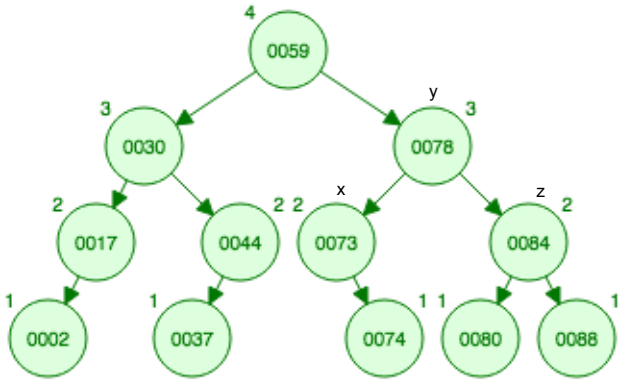
iii) AVL tree after remove(70) is performed with time complexity O(log(n)):



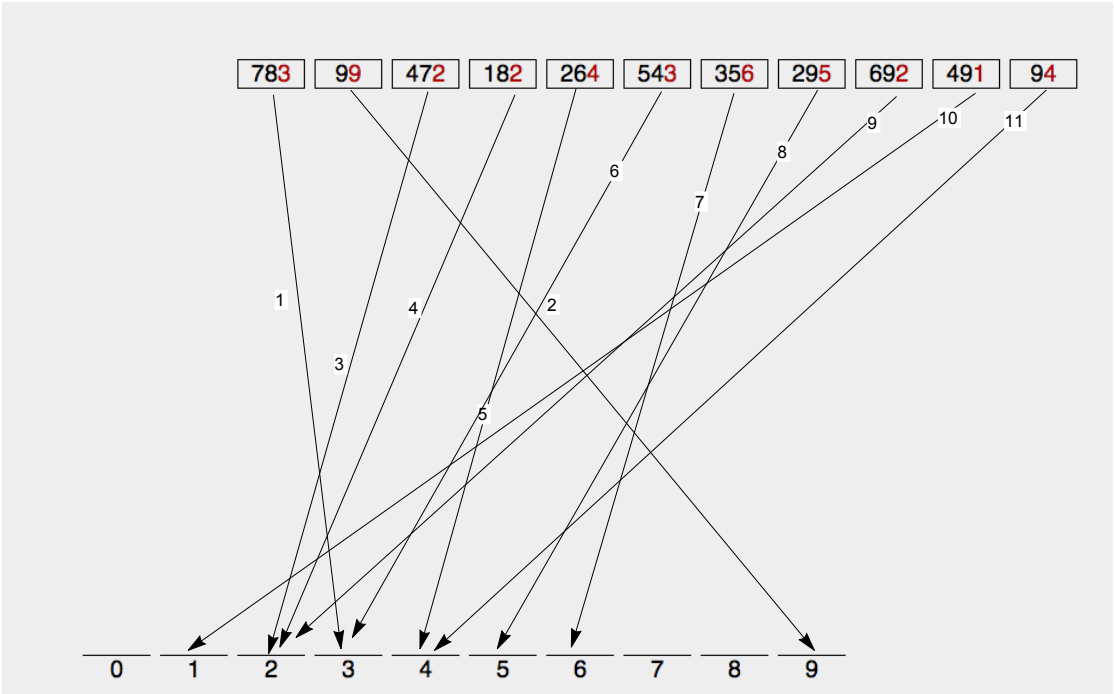


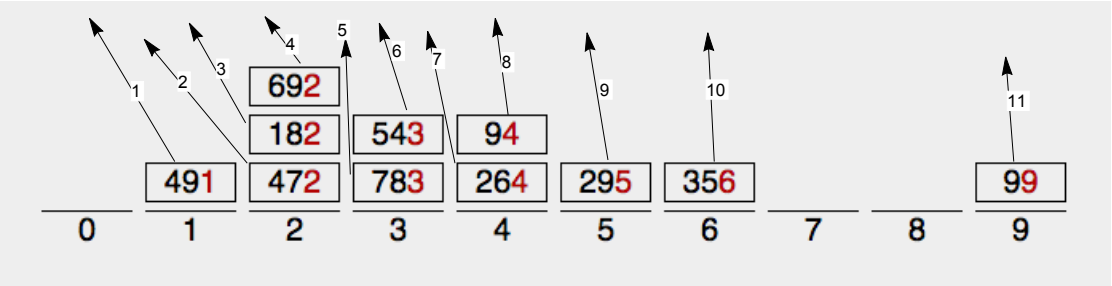
iv) AVL tree after remove(91) is performed and the worst running time is O(log n):

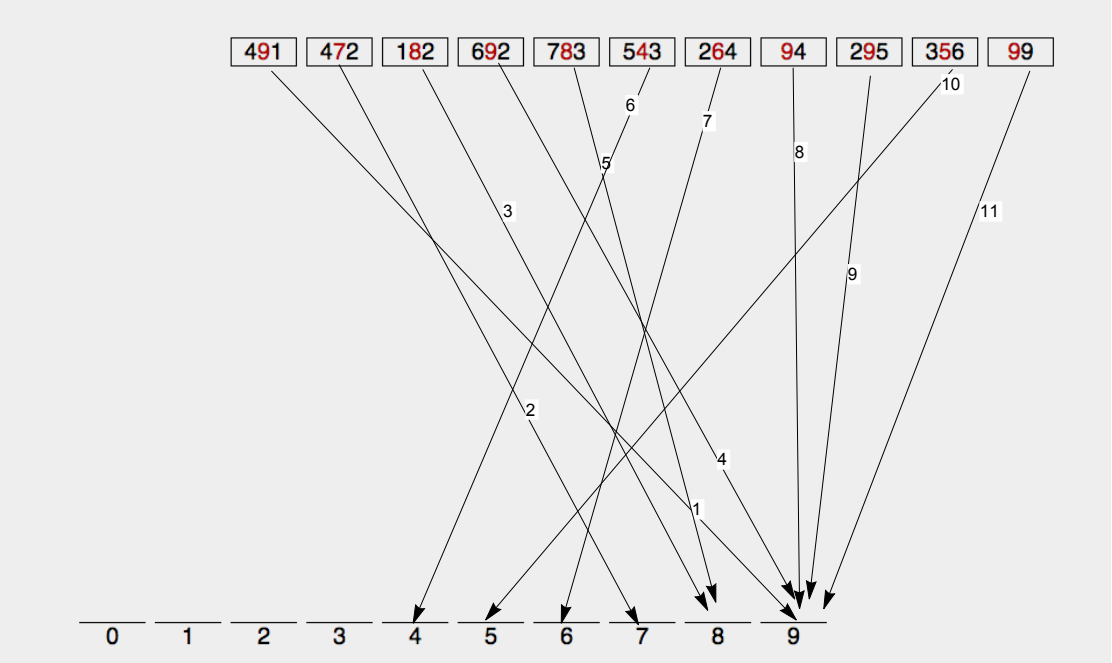


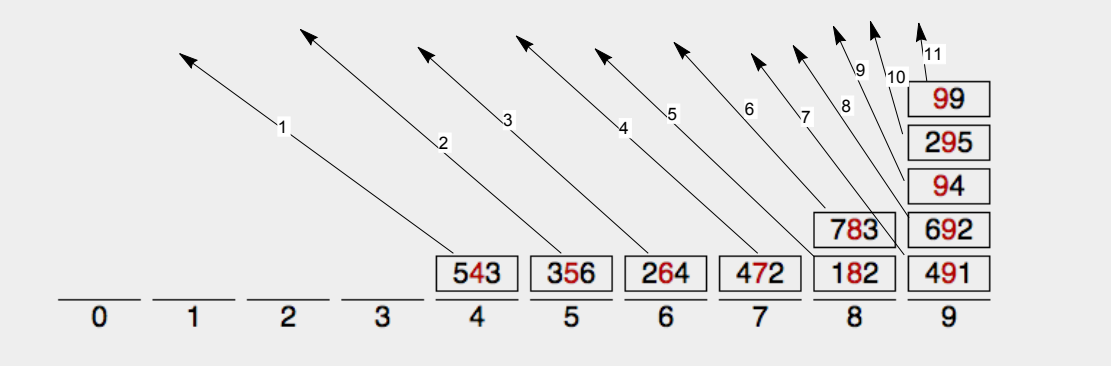

4
0059
3
0030
y
3
0078
2
0017
2
0044
2
x
2
0073
z
2
0084
1
0002
1
0037
1
0074
1
0080
1
0088

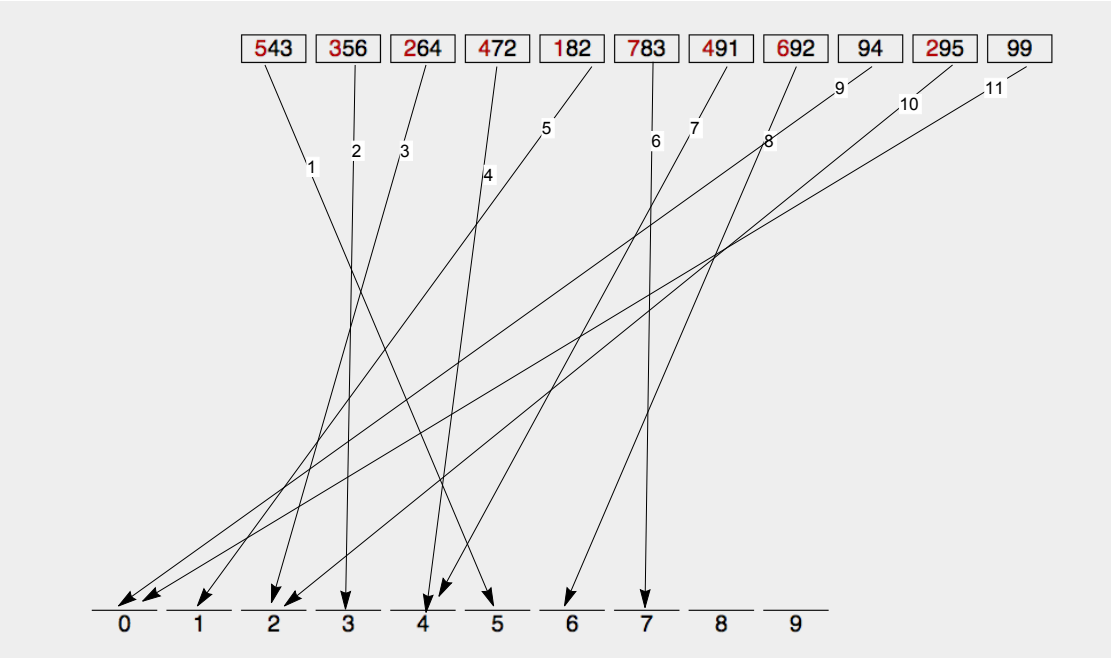
Question 12

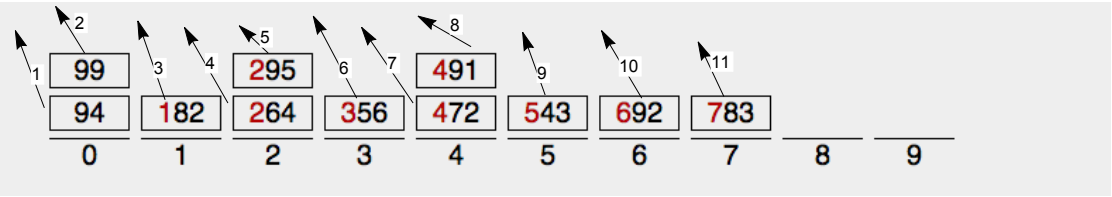

783
99
472
182
264
543
356
295
692
491
94
1
2
3
4
5
6
7
8
9
10
11
0
1
2
3
4
5
6
7
8
9


692
182
543
94
491
472
783
264
295
356
99
0 1 2 3 4 5 6 7 8 9


491 472 182 692 783 543 264 94 295 356 99
0 1 2 3 4 5 6 7 8 9


99
295
94
783
692
543
356
264
472
182
491
0 1 2 3 4 5 6 7 8 9





94 99 182 264 295 356 472 491 543 692 783