Mohammad Ali Zahir

COMP 352: Data Structures and Algorithms

18th April 2020

Assignment 4

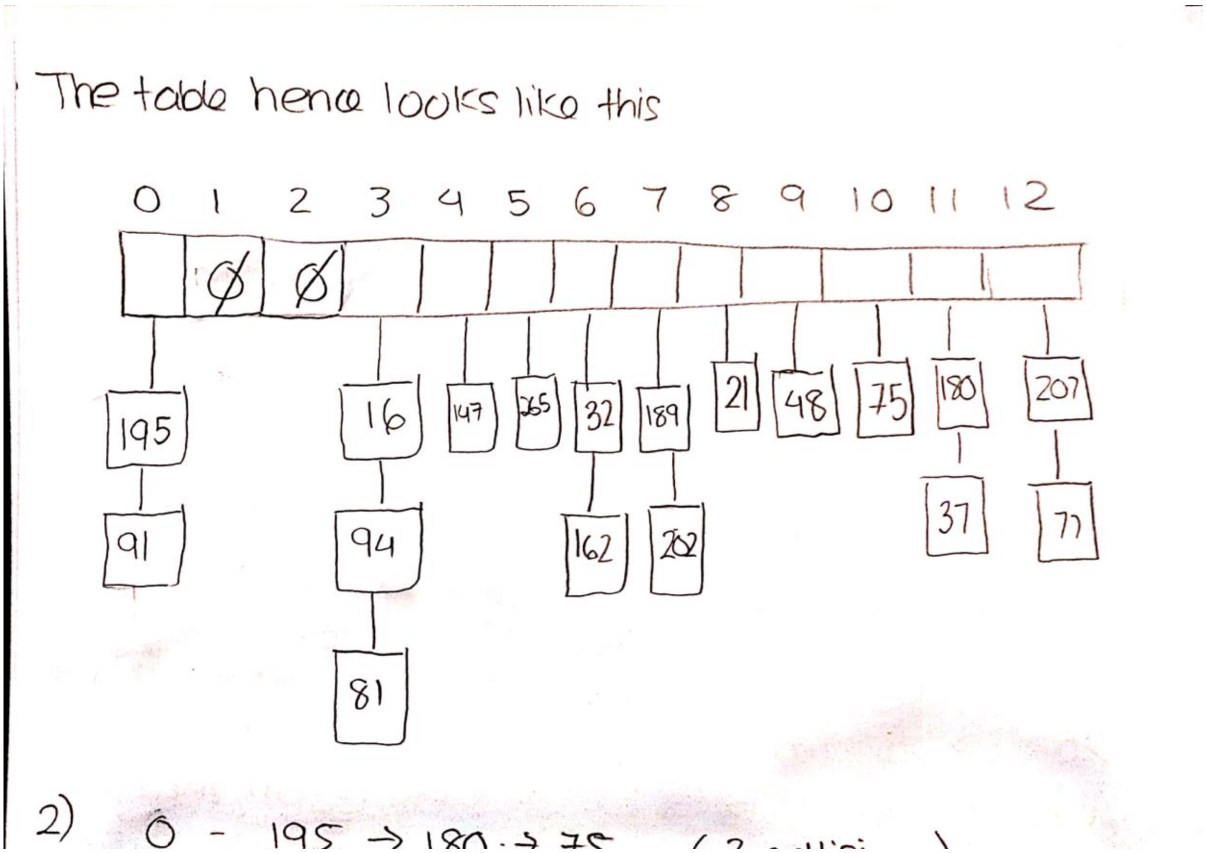
1. i)

1) 32, 147, 265, 195, 207, 180, 21, 16, 189, 202, 91, 94, 162
i) 75, 37, 77, 81, 48

Hash function: $h(k) = k \bmod 13$

$k=32 \to 32 \bmod 13 \to 6$
$k=147 \to 147 \bmod 13 \to 4$
$k=265 \to 265 \bmod 13 \to 5$
$k=195 \to 195 \bmod 13 \to 0$
$k=207 \to 207 \bmod 13 \to 12$
$k=180 \to 180 \bmod 13 \to 11$
$k=21 \to 21 \bmod 13 \to 8$
$k=16 \to 16 \bmod 13 \to 3$
$k=189 \to 189 \bmod 13 \to 7$
$k=202 \to 202 \bmod 13 \to 7$
$k=91 \to 91 \bmod 13 \to 0$
$k=94 \to 94 \bmod 13 \to 3$
$k=162 \to 162 \bmod 13 \to 6$
$k=75 \to 75 \bmod 13 \to 10$
$k=37 \to 37 \bmod 13 \to 11$
$k=77 \to 77 \bmod 13 \to 12$
$k=81 \to 81 \bmod 13 \to 3$
$k=48 \to 48 \bmod 13 \to 9$

Table is on the other side.



ii) At one index, there are at most 2 collisions, but in total there are 7 collisions.

2. This proposal would not be valid here since we would have more collisions than question 1 (see picture below). However, the idea of this practice makes sense since if we increase the table size, the better chance we have to cause less collisions due to the table having more cells to insert elements in. We can just say that the data was not optimal here.

$\boxed{81}$

2) 0 - 195 → 180 → 75 (2 collisions)

1 - 16 → 91 (1 collision)

2 - 32 → 77 (1 collision)

3 - 48

4 - 94

5 - ∅

6 - 21 → 81 (1 collision)

7 - 202 → 37 (1 collision)

8 - ∅

9 - 189

10 - 265

11 - ∅

12 - 147 → 207 → 162 (2 collisions)

13 - ∅

14 - ∅

We would have a total of 8 collisions

3. i)

3)
0 -
1 - 39
2 -
3 - 29
4 - 42
5 -
6 - ~~25~~ 35
7 -
8 -
9 -
10 - 48
11 - 35
12 - 12
13
14 -
15 -
16 - ~~31~~ 29
17 -
18 - 18

/ = remove

Rest of the operations on the other side.

put(25) = 25 mod 19 = 6

put(12) = 12 % 19 = 12

put(42) = 42 % 19 = 4

put(31) = 31 % 19 = 12

There is a collision

7 - 31 mod 7 = 4

12 + 1(4) = 16

put(35)

35 mod 19 = 16

There is a collision

7 - 35 mod 7 = 7

16 + 2(7) mod 19 = 11

put(39) → 39 mod 19 = 1

remove(31) → 31 mod 19 = 12 , it's not there

7 − 31 mod 7 = 4

12 + (1)(4) = 16 → This is true, remove at 16

put(48) = 48 mod 19 = 10

remove(25) = 25 mod 19 = 6 ⇒ This is true, remove at 6

put(18) = 18 mod 19 = 18

put(29) = 29 mod 19 = 10 There is a collision

7 − 29 mod 7 = 6

10 + 1(6) = 16

put(29) → collision at 10

10 + 1(6) = 16

10 + 2(6) = 22 → 22 mod 19 = 3

put(35) → Collision at 16   7 − 35 mod 7 =

16 + 1(7) = 23 mod 19 = 4 X

16 + 2(7) = 30 mod 19 = 11 X

16 + 3(7) = 37 mod 19 = 18 X

16 + 4(7) = 44 mod 19 = 6

Hence, the updated table would look like

0 - Ø
1 - 39
2 - Ø
3 - 29
4 - 42
5 - Ø
6 - 35
7 - Ø
8 - Ø
9 - Ø
10 - 48
11 - 35
12 - 12
13 - Ø
14 - Ø
15 - Ø
16 - 29
17 - Ø
18 - 18

ii) The size of the longest cluster would be 3 at the indexes 10-11-12

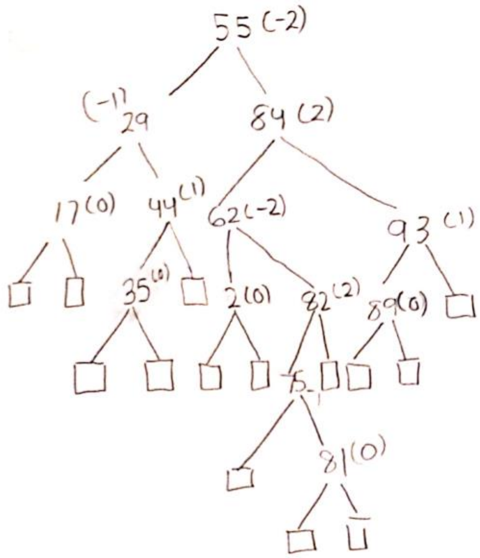
iii) There was a total of 10 collisions

iv) The load factor is 9/19 or 47.37%

4.

4)

i)



A (b)

balance factor

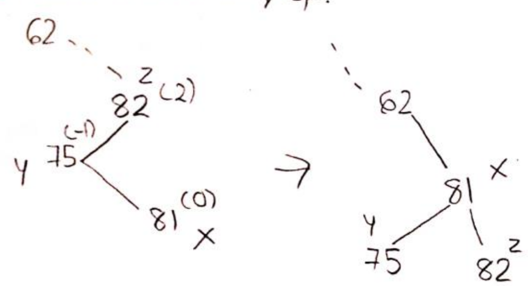
↓

hright-hleft

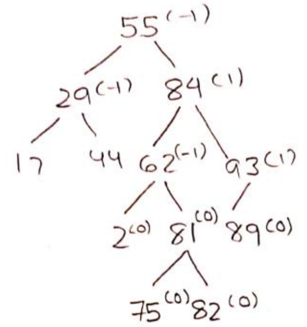
Hence, here there are errors with 55, 84, 62, 82

To correct this we use 82 (the farthest root)

and work on our way up.

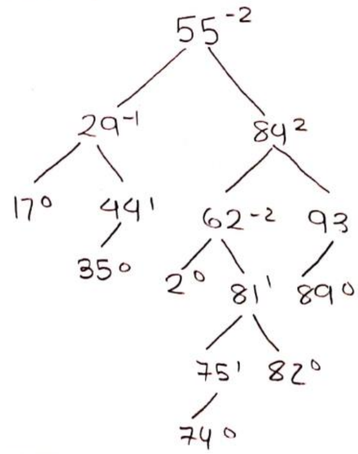


The whole tree would then look like



→ This is a balanced tree.

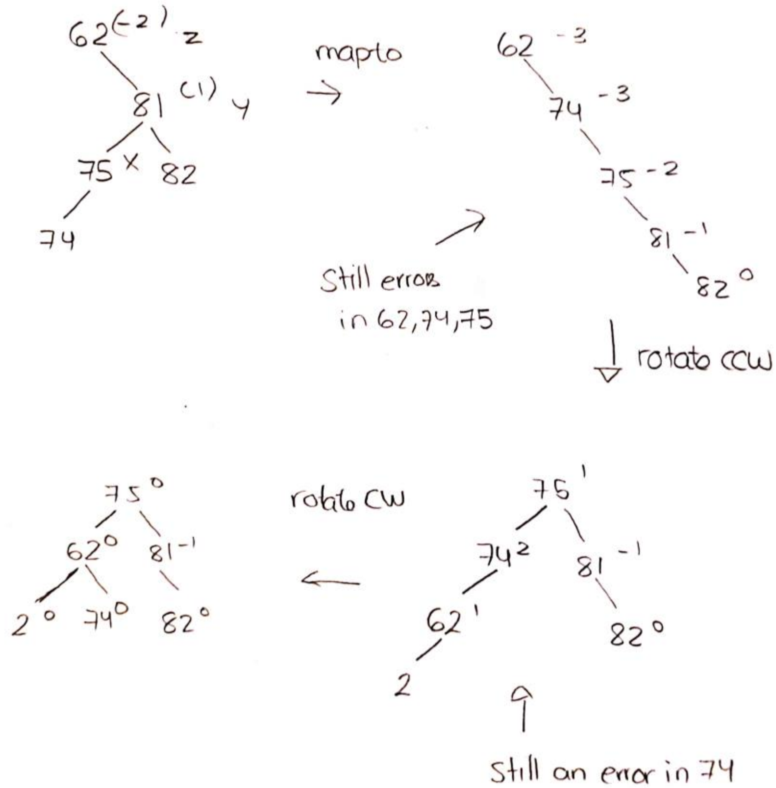
ii) put(74)



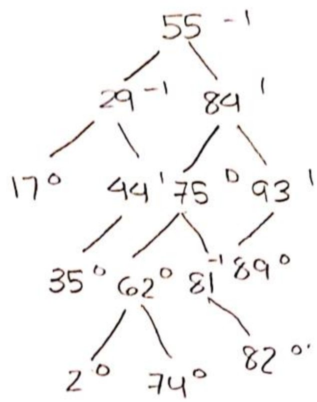
Erros at 55, 84, 62

iii

We take 62 since it is farthest from the right.



Hence the entire tree would look like

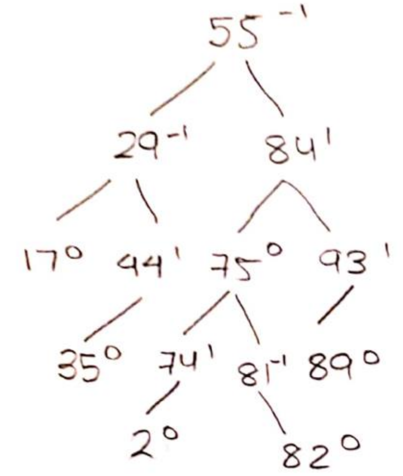


Since the tree is now balanced, we know that the complexity is O(h), which in this case is O(log n)

iii) remove(62), This has two children, hence we replace the node with the predecessor (parent)

Hence, we replace 62 with 75

Hence, tree would look like



→ This is a balanced tree, no error

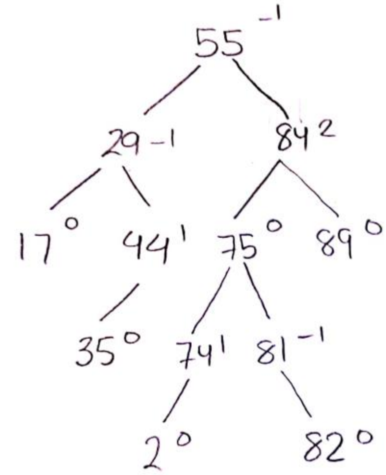
Hence, complexity is O(logn).

iv) remove(93). Only has one child.

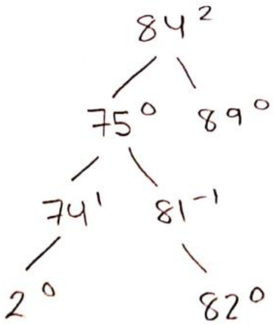
Hence we have to connect it's parent and child.

Connect 84 to 89

Hence the tree would look like,

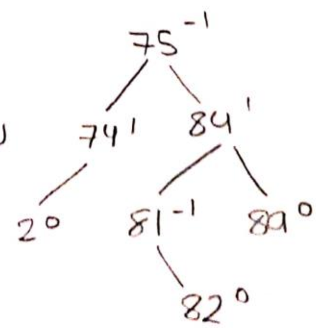


Error in 84

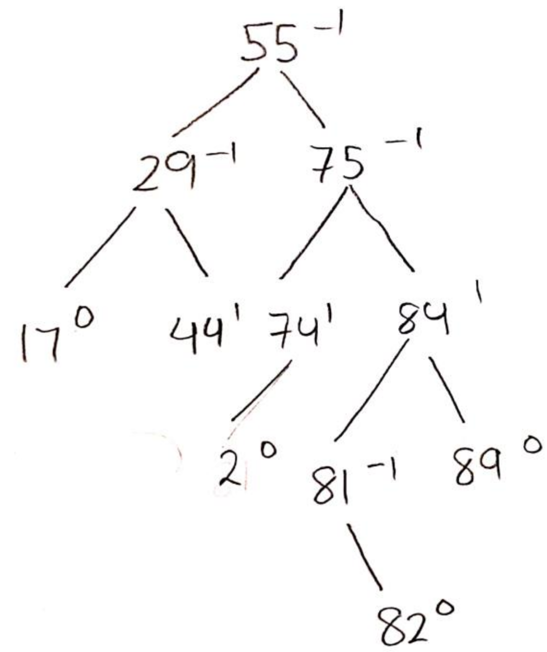


Left-side imbalance.

→ rotate cw



Hence, the final tree looks like



→ Balanced tree

Hence, complexity

is O(logn).

5)

5) 832, 91, 411, 172, 243, 573, 326, 292, 682, 489
96

At most, there are 3 digits in one number

If we sort by the first digit,

91 411 832 172 292 682 243 573 326
96 489

If we sort by the second digit,

411 326 832 243 172 573 682 489 91
292 96

If we sort by the third digit,

91 96 172 243 292 326 411 489 573
682 832

This is the array sorted by the radix sort