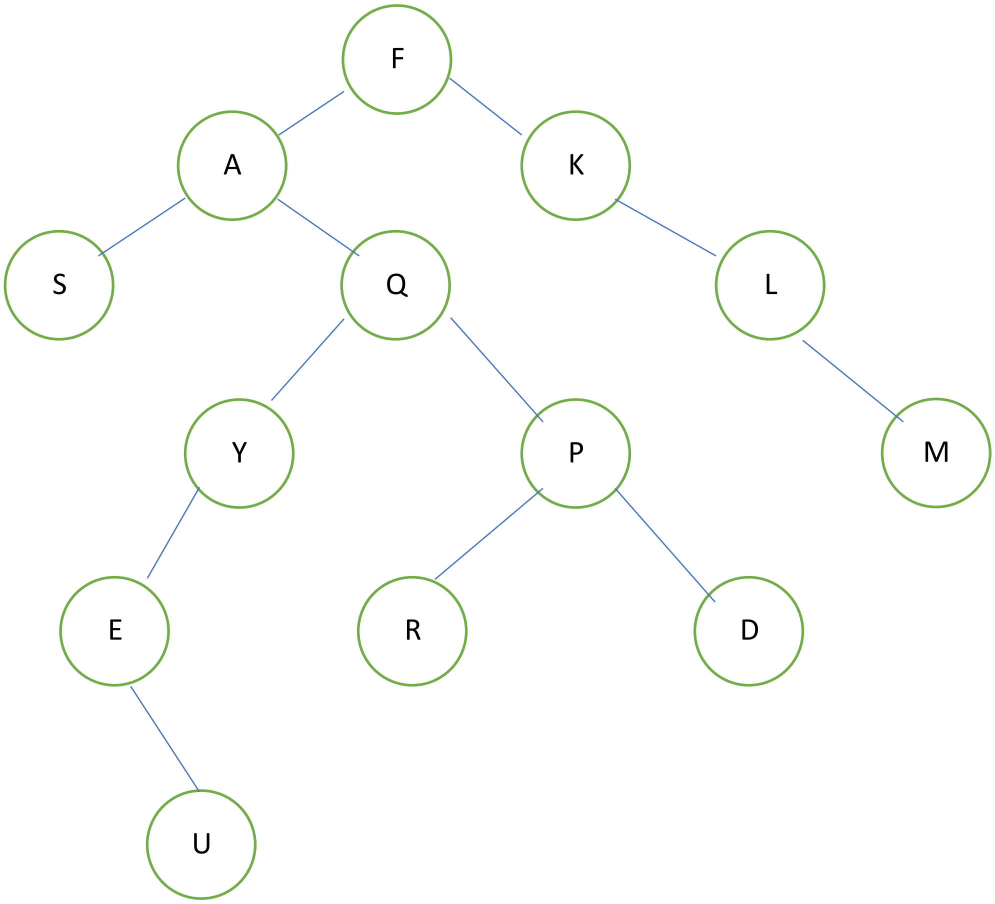
Zafir Khalid - 40152164

Question 1

Draw a single binary tree that gave the following traversals:

Inorder: S A E U Y Q R P D F K L M

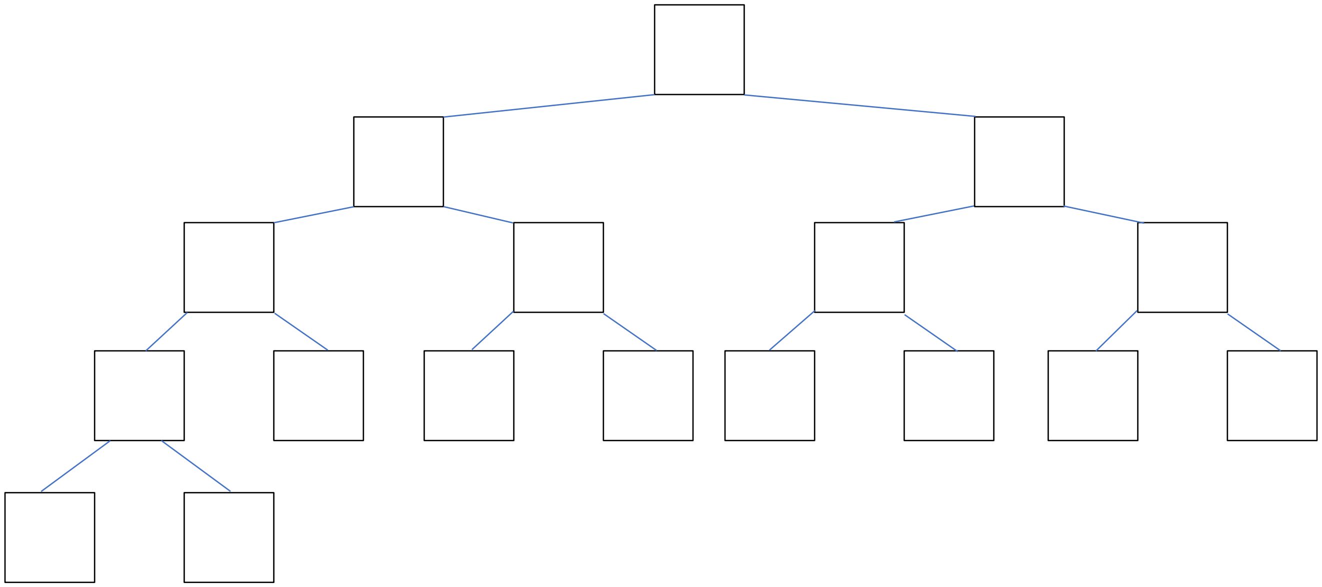
Preorder: F A S Q Y E U P R D K L M



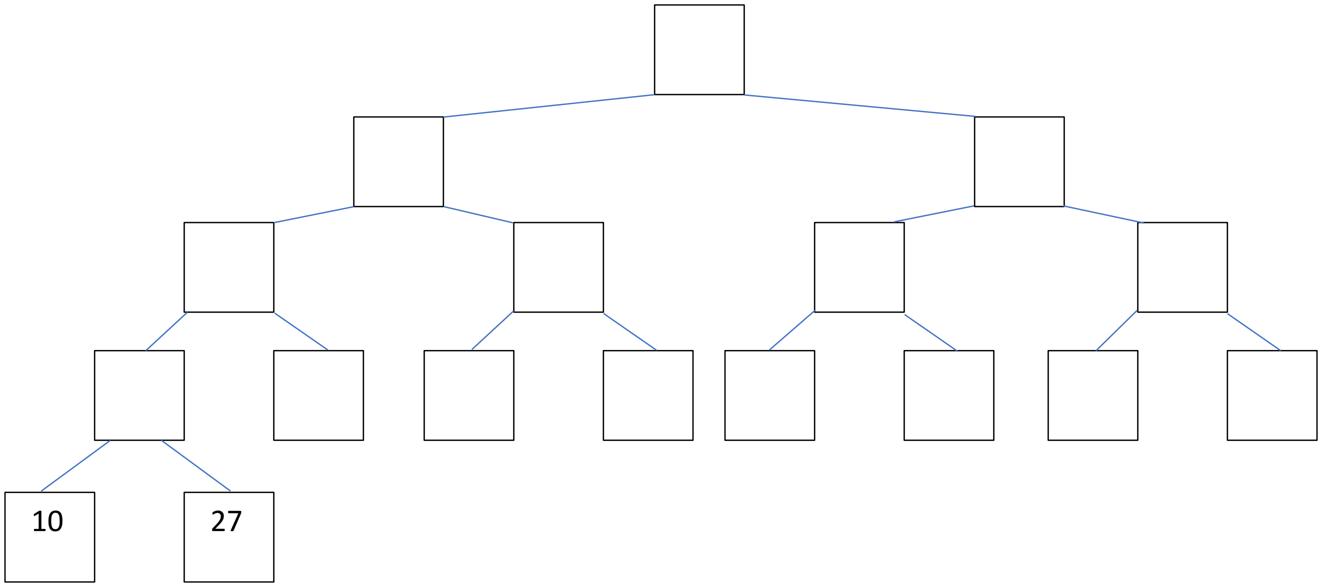
| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| F | A | K | S | Q | | L | | | Y | P | | | | M | | | | | E |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| | R | D | | | | | | | | | | | | | | | | | |
| 40 | | | | | | | | | | | | | | | | | | | |
| U | | | | | | | | | | | | | | | | | | | |

Question 2
Draw the min-heap that results from the bottom-up heap construction algorithm on the following list of values:
10, 27, 15, 35, 42, 19, 45, 16, 12, 8, 18, 14, 13, 9, 20, 11, 13
Starting from the bottom layer, use the values from left to right as specified above. Show all the steps and the final tree representing the min-heap. Afterwards perform the operation removeMin four (4) times and show the resulting min-heap after each step.

#1 Create Empty Tree

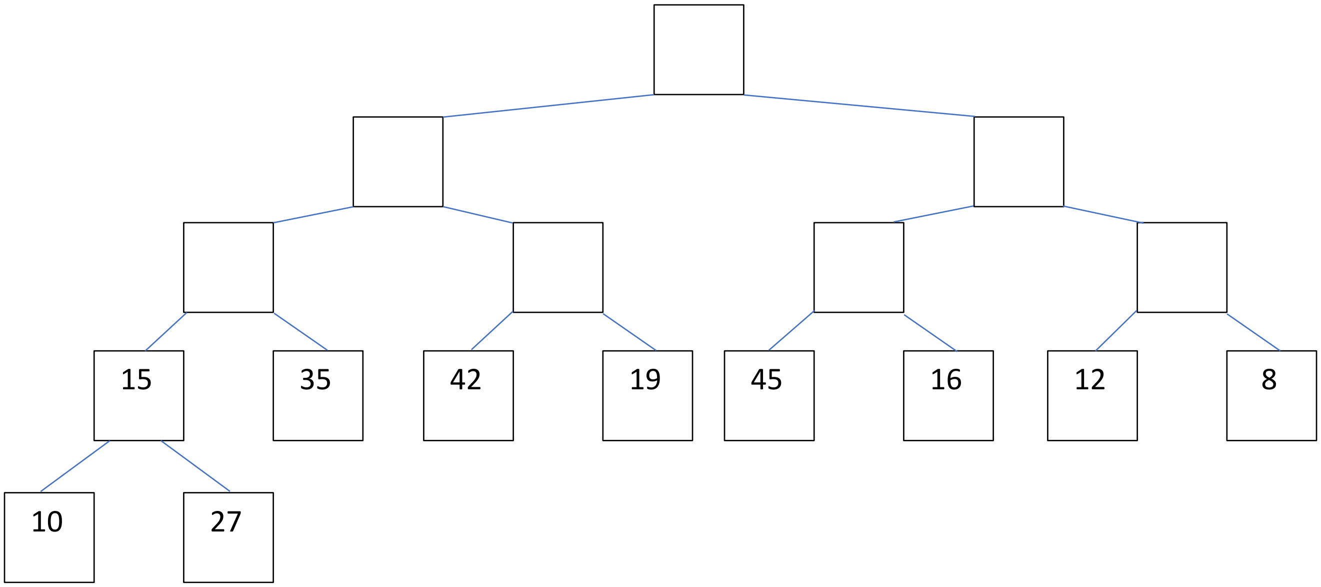


#2 Add from bottom

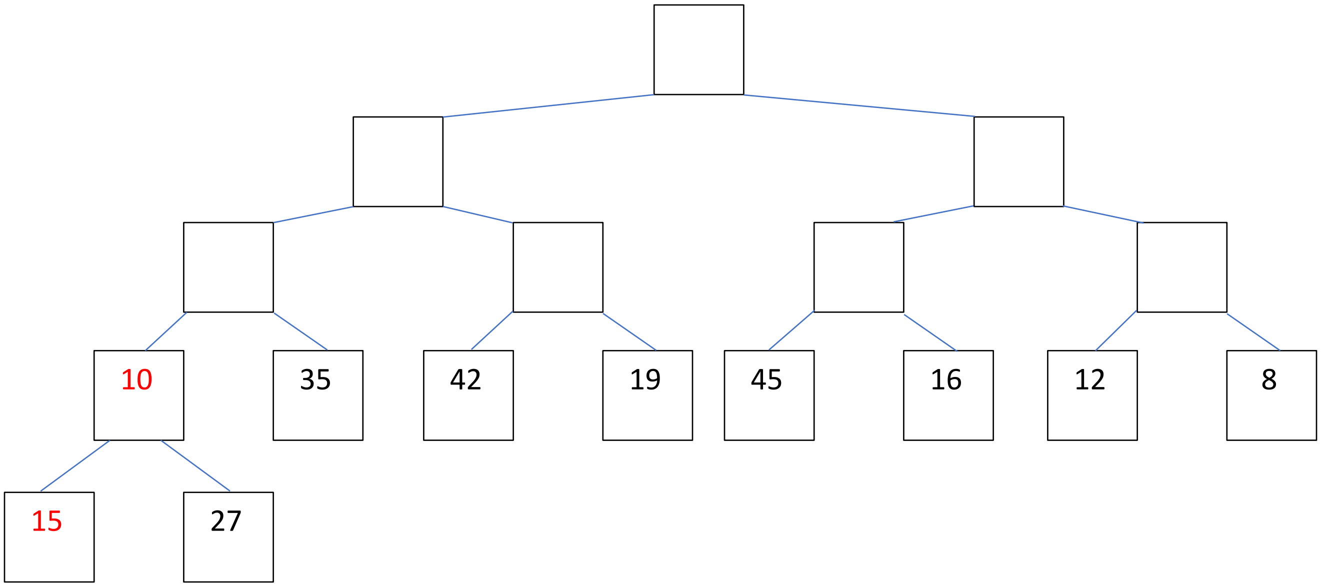


Zafir Khalid - 40152164

#3 Add next row

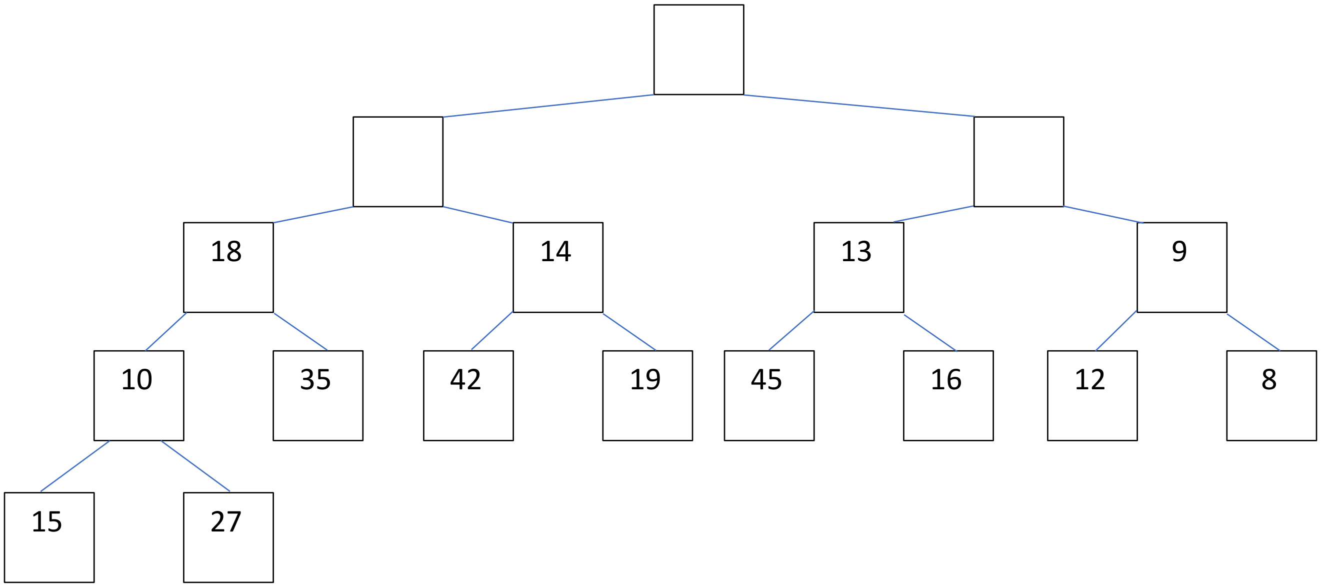


#4 Bubble up

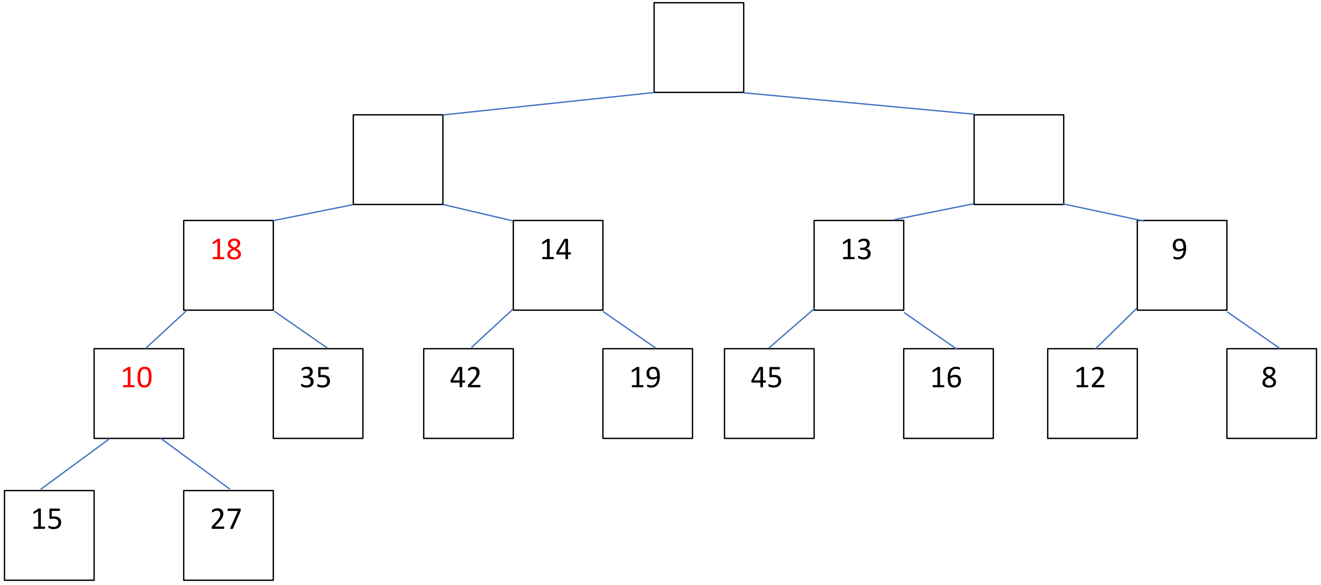


Zafir Khalid - 40152164

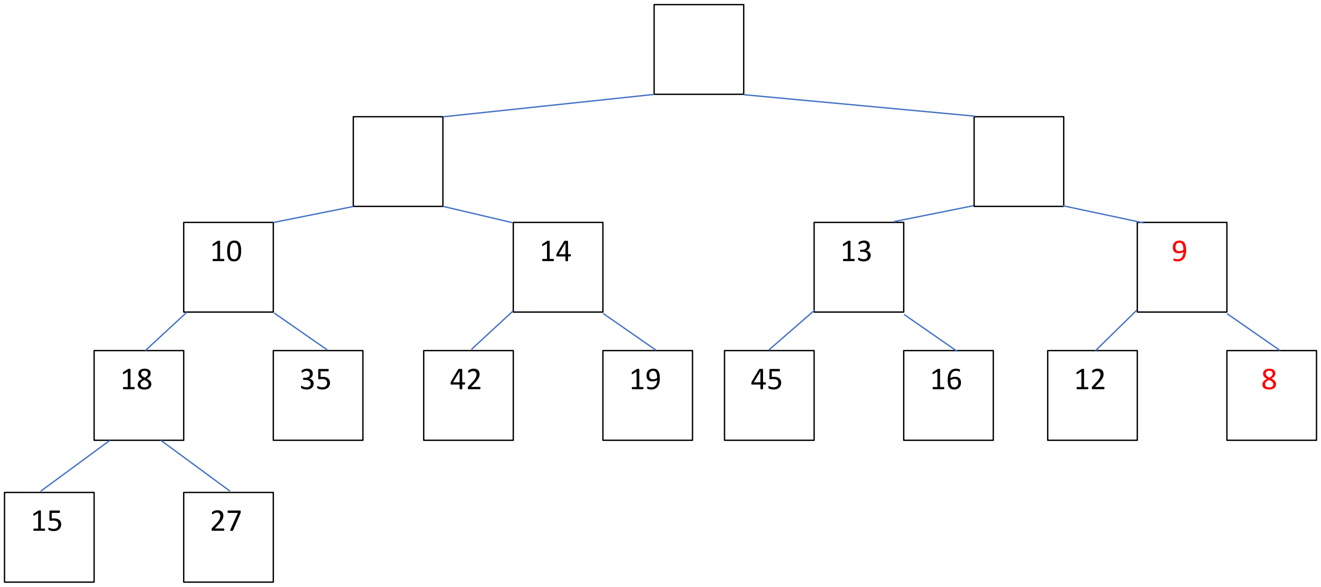
#5 Add next row



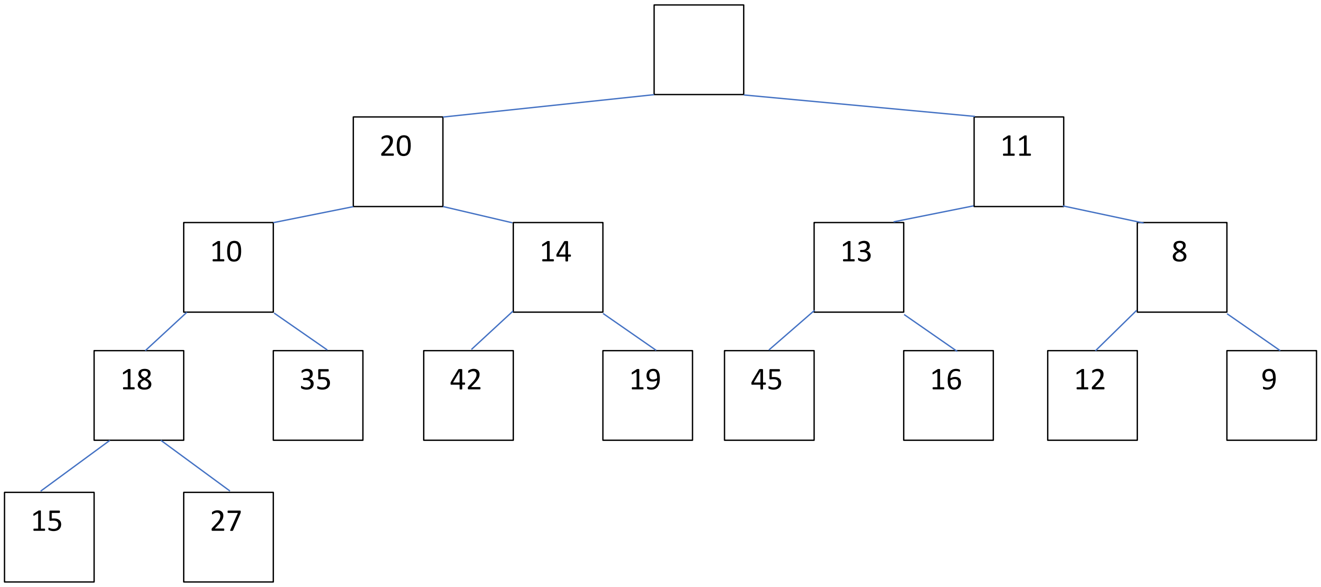
#6 Bubble up



#7 Bubble up

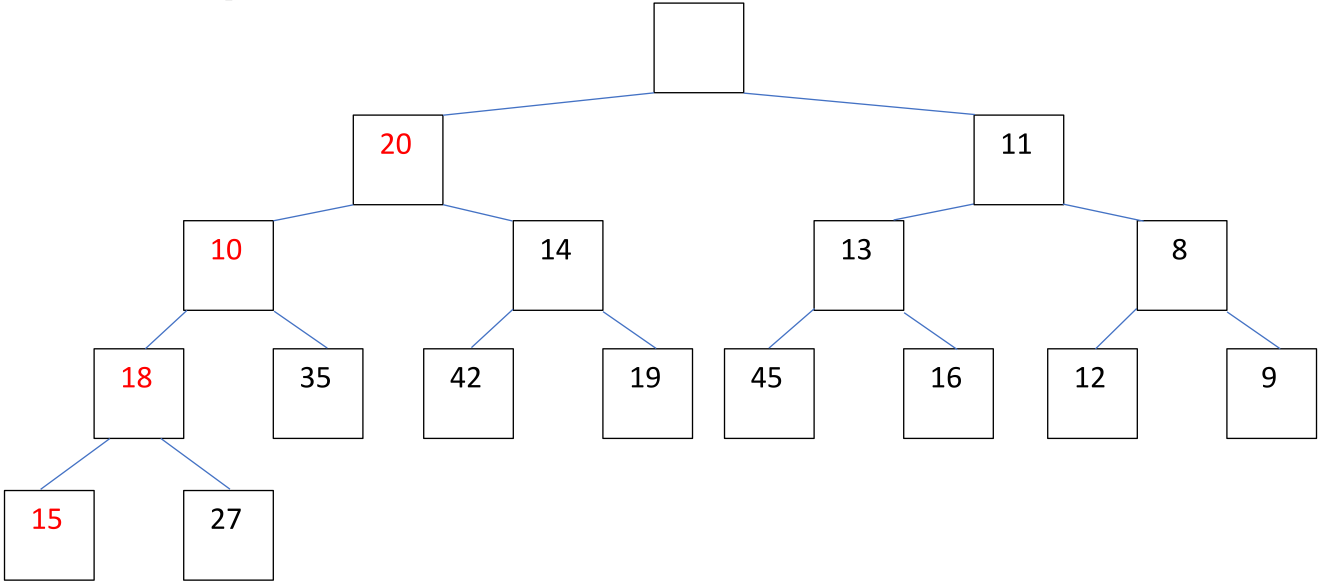


#8 Add next row

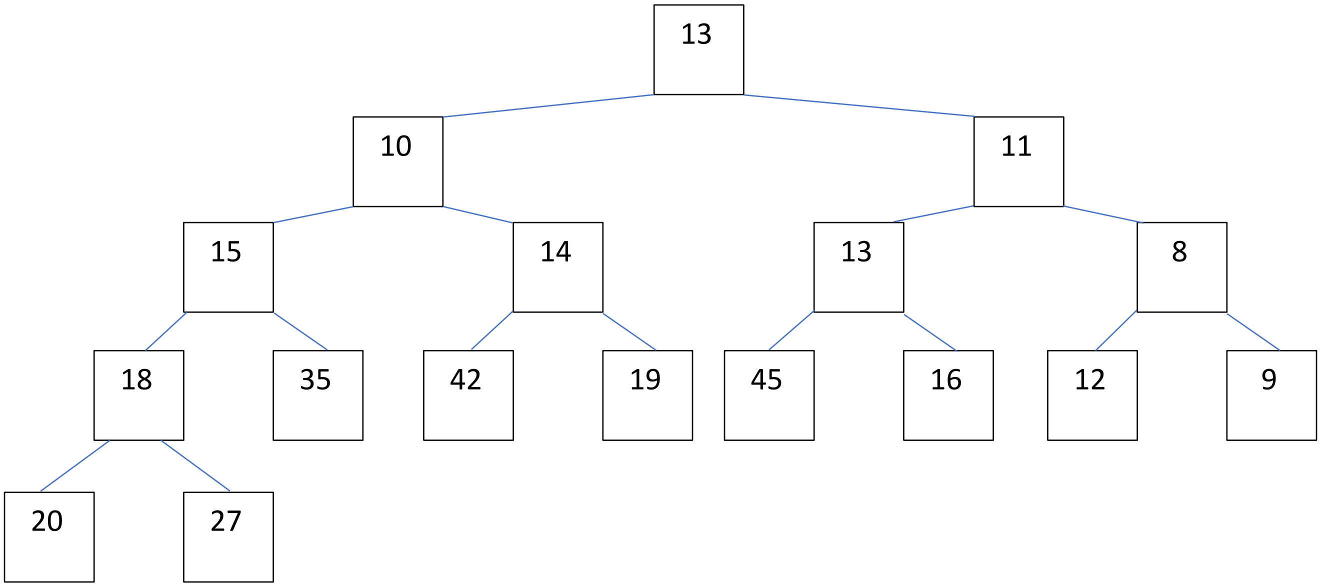


Zafir Khalid - 40152164

#9 Bubble up

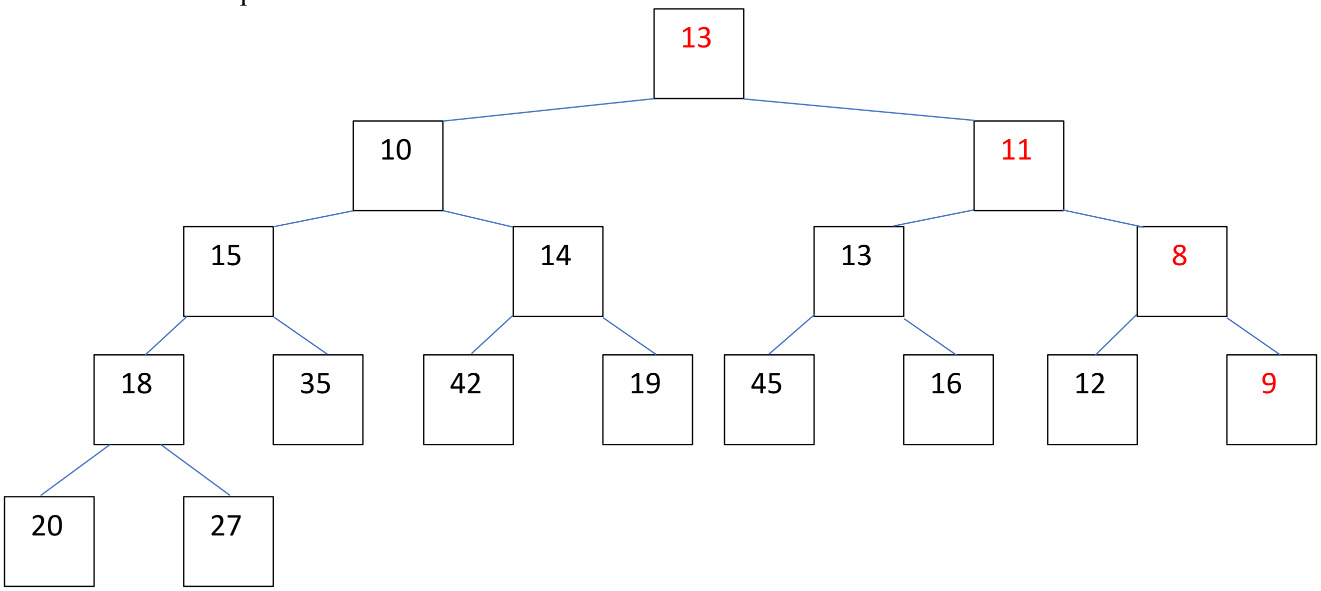


#10 Add root

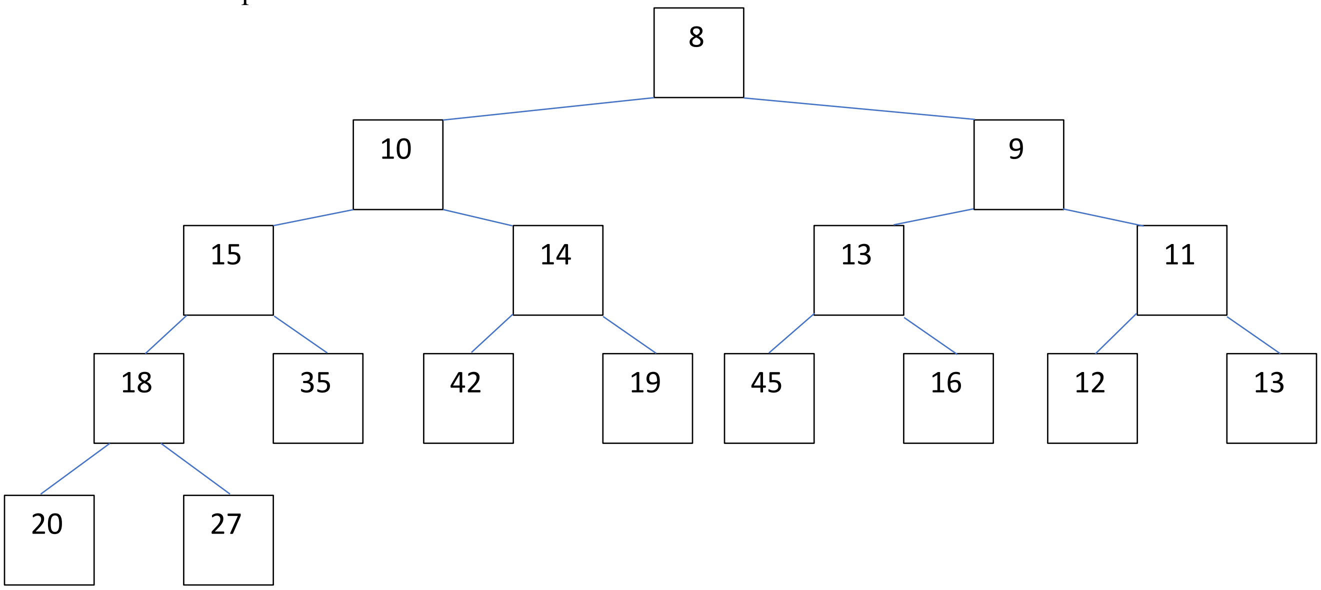


Zafir Khalid - 40152164

#11 Bubble up

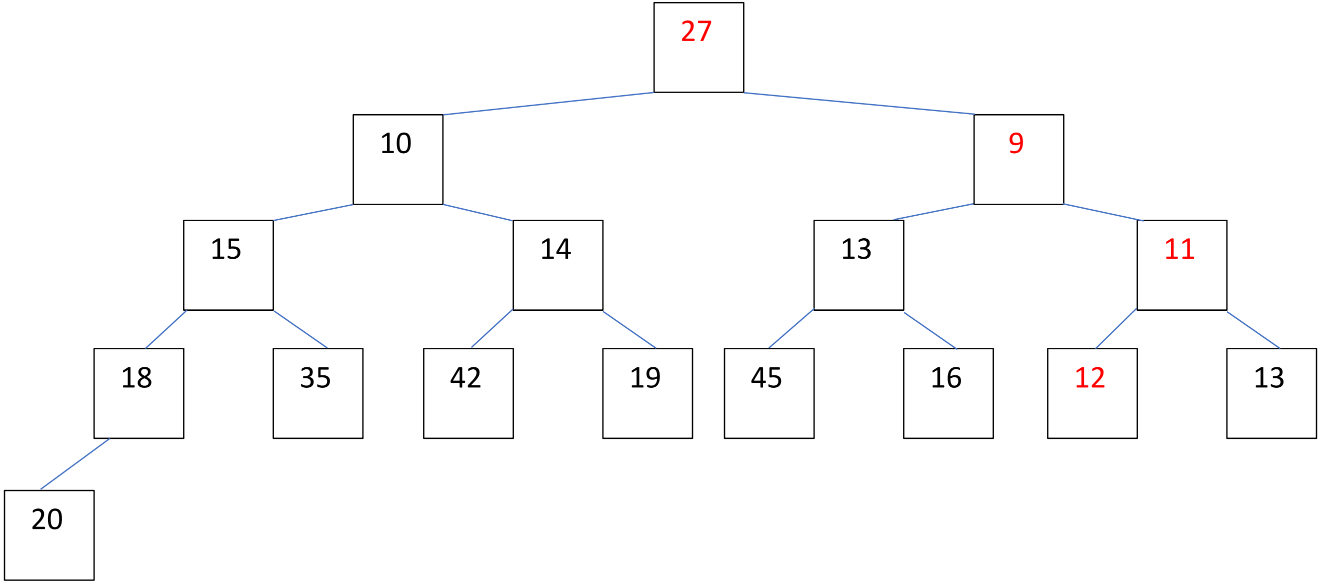


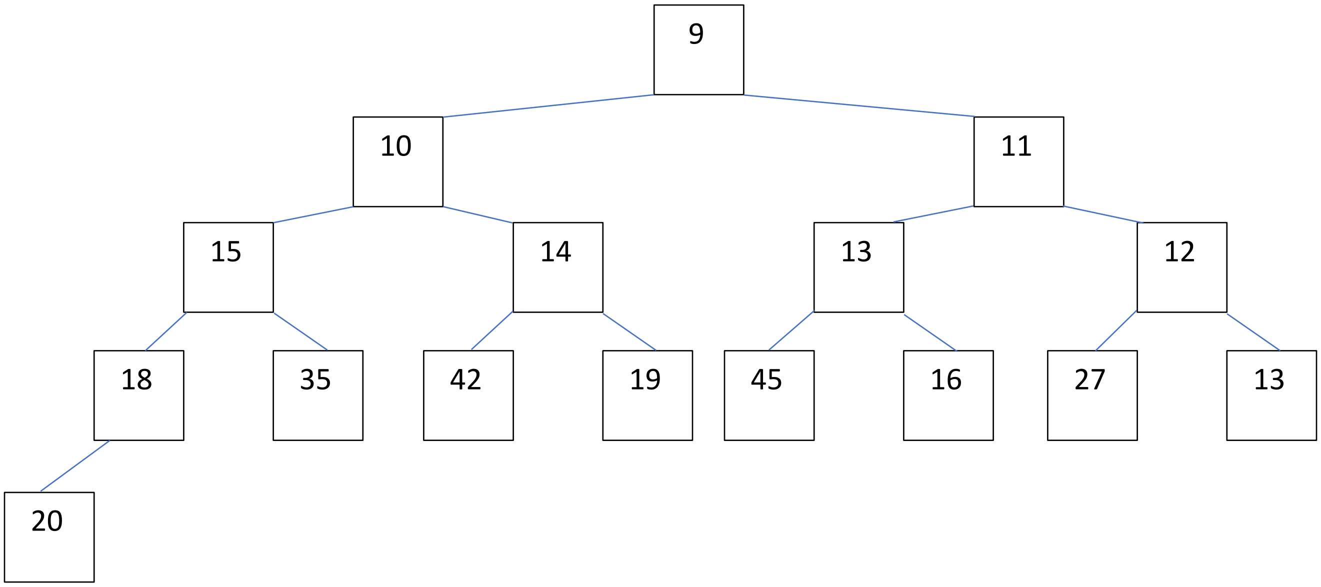
#12 Final heap



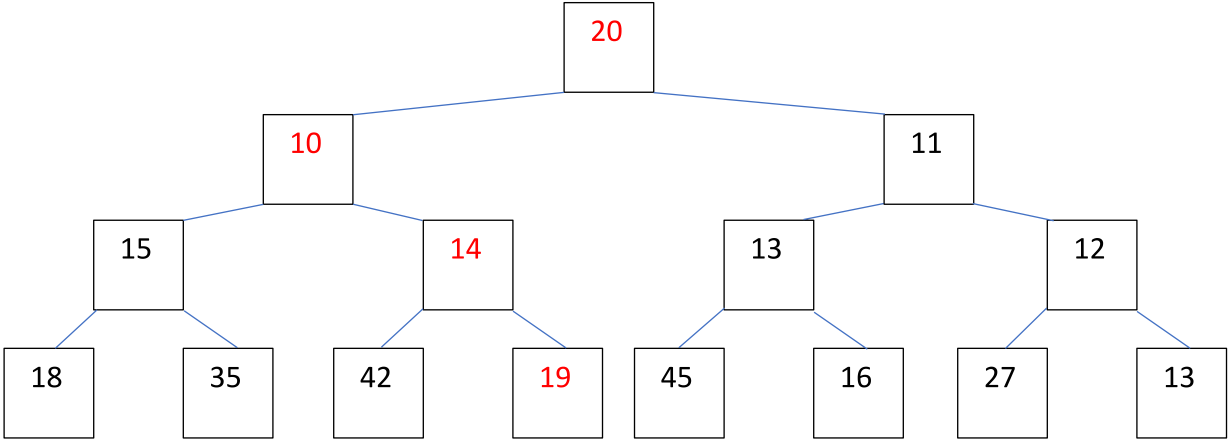
Zafir Khalid - 40152164

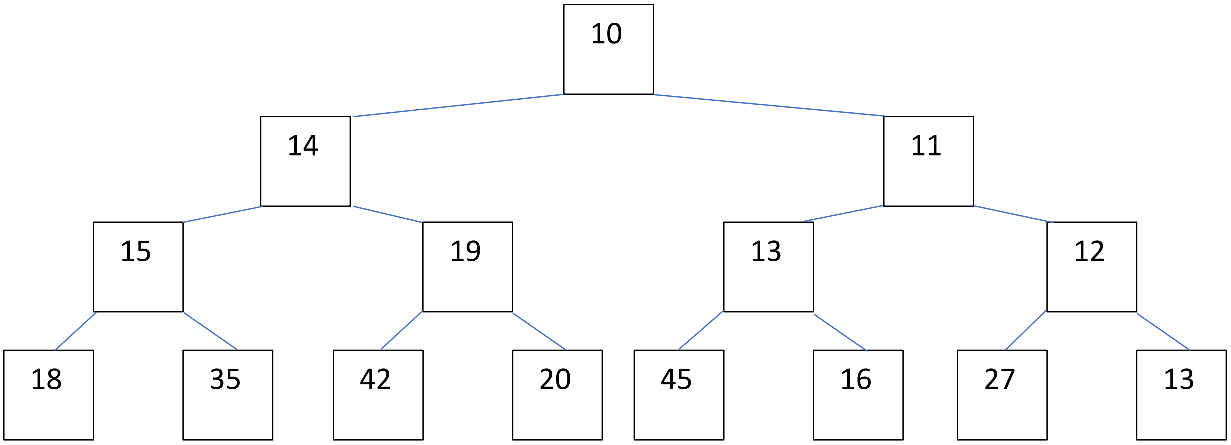
1. removeMin()



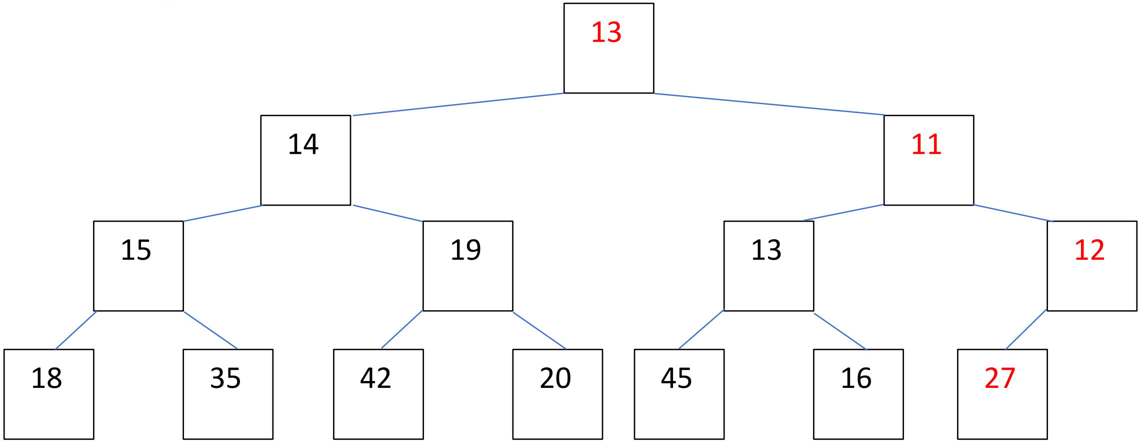


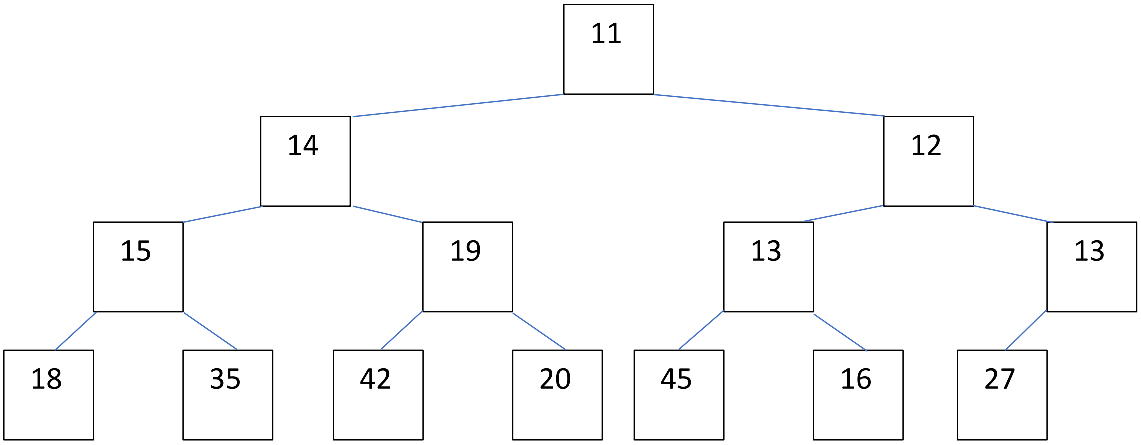
2. removeMin()


20
10
11
15
14
13
12
18
35
42
19
45
16
27
13

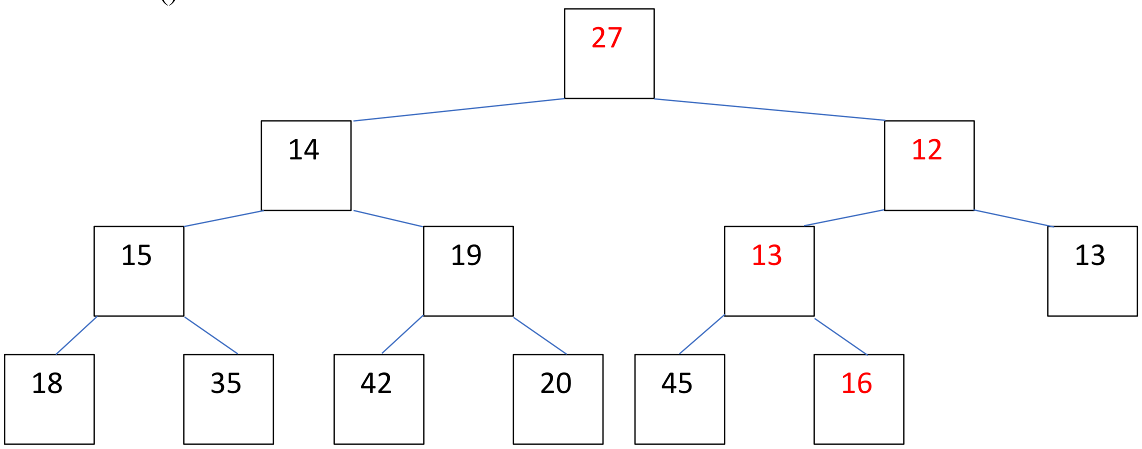

10
14
11
15
19
13
12
18
35
42
20
45
16
27
13

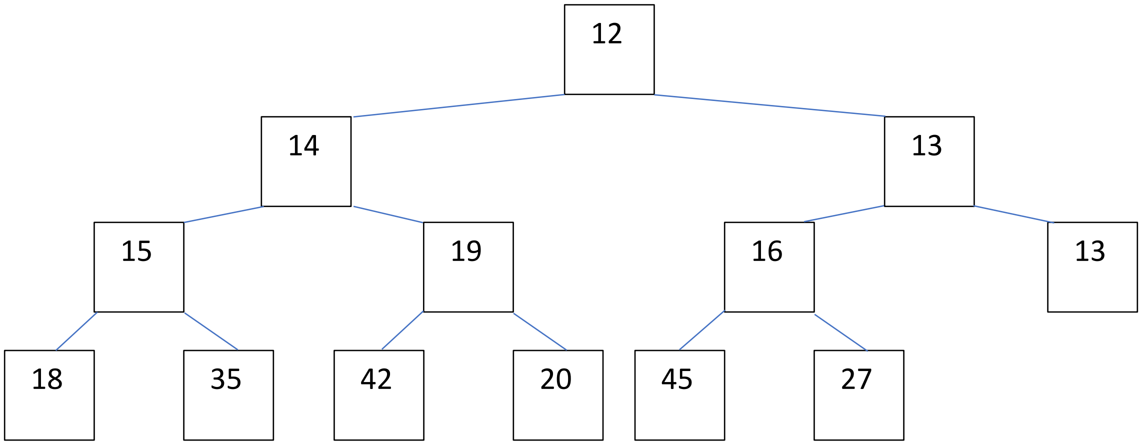
3. removeMin()


13
14
11
15
19
13
12
18
35
42
20
45
16
27


11
14
12
15
19
13
13
18
35
42
20
45
16
27

4. removeMin()


27
14
12
15
19
13
13
18
35
42
20
45
16

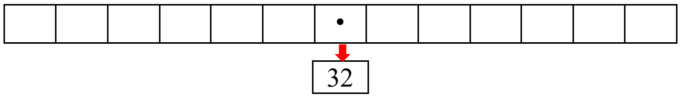

12
14
13
15
19
16
13
18
35
42
20
45
27

Question 3

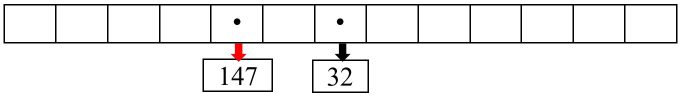
Assume a hash table that utilizes an array of 13 elements and where collisions are handled by separate chaining. Considering the hash function is defined as: h(k)=k mod 13.

a) Draw the contents of the table after inserting elements with the following keys:
{32, 147, 265, 195, 207, 180, 21, 16, 189, 202, 91, 94, 162, 75, 37, 77, 81, 48}

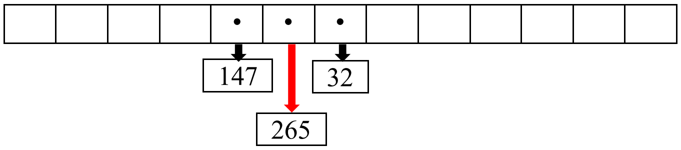
Insert 32 h(32) = 32 mod 13 = 6



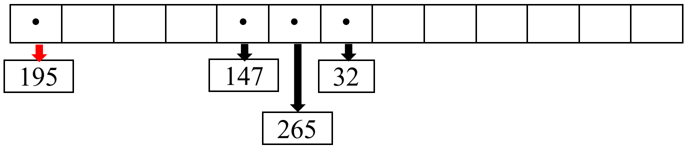
Insert 147 h(147) = 147 mod 13 = 4



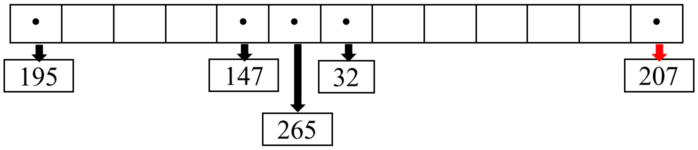
Insert 265 h(265) = 265 mod 13 = 5



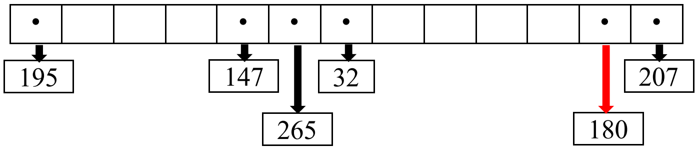
Insert 195 h(195) = 195 mod 13 = 0



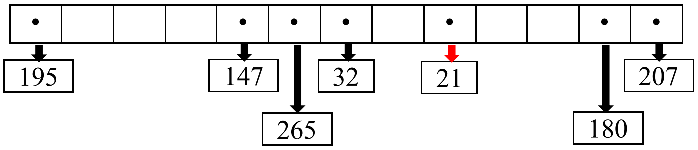
Insert 207 h(207) = 207 mod 13 = 12



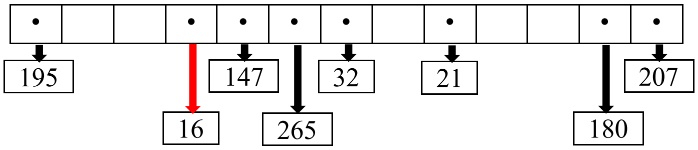
Insert 180 h(180) = 180 mod 13 = 11



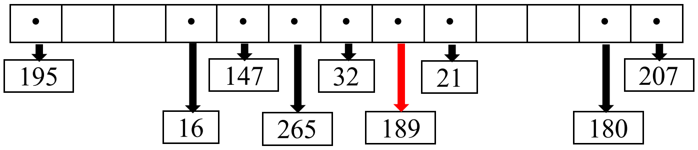
Insert 21 h(21) = 21 mod 13 = 8



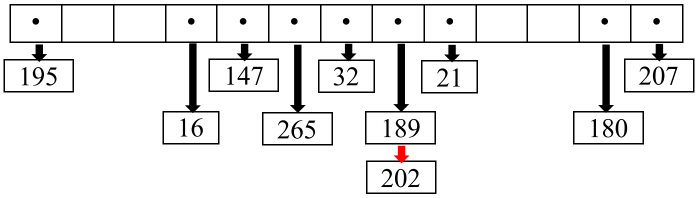
Insert 16 h(16) = 16 mod 13 = 3



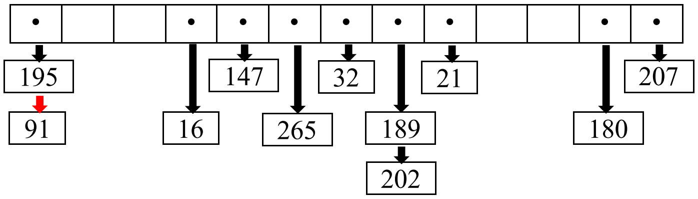
Insert 189 h(189) = 189 mod 13 = 7



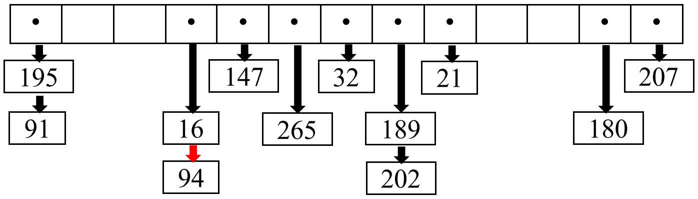
Insert 202 h(202) = 202 mod 13 = 7 [Collision #1]



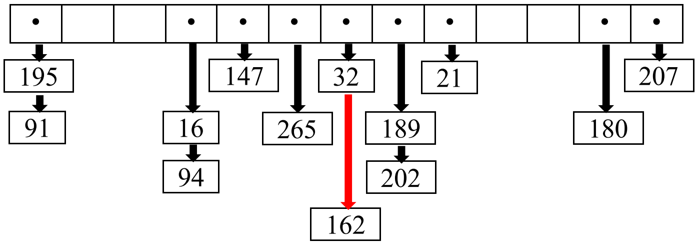
Insert 91 h(91) = 91 mod 13 = 0 [Collision #2]



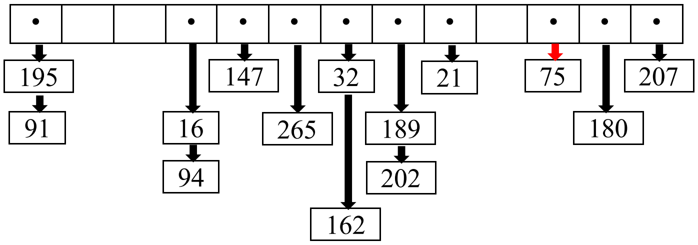
Insert 94 h(94) = 94 mod 13 = 3 [Collision #3]



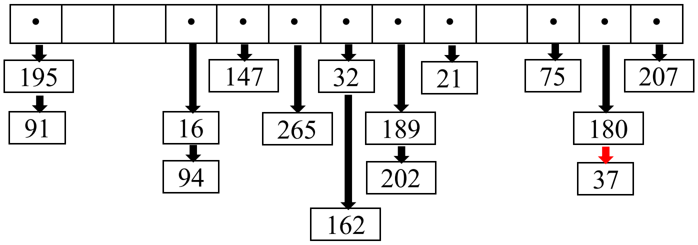
Insert 162 h(162) = 162 mod 13 = 6 [Collision #4]



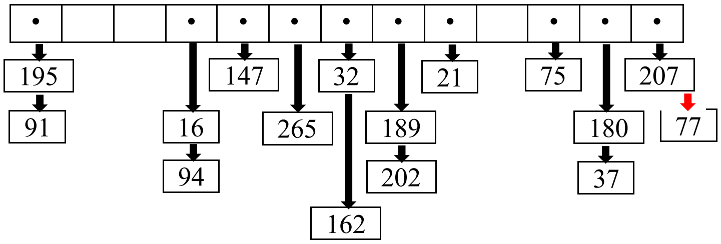
Insert 75 h(75) = 75 mod 13 = 10



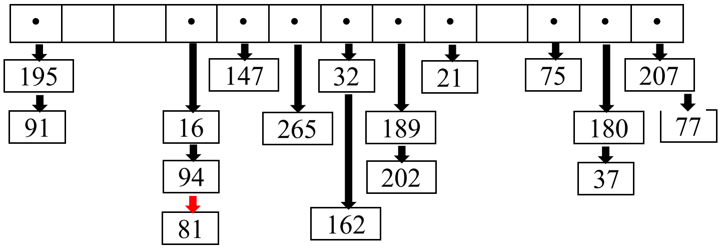
Insert 37 h(37) = 37 mod 13 = 11 [Collision #5]



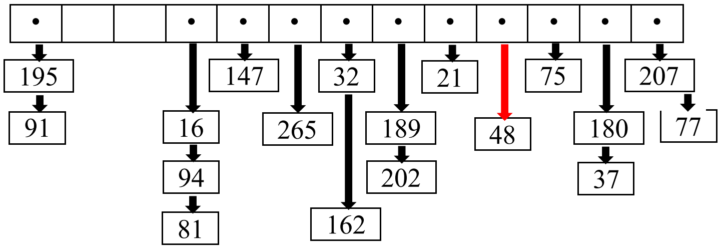
Insert 77 h(77) = 77 mod 13 = 12 [Collision #6]



Insert 81 h(81) = 81 mod 13 = 3 [Collision #7]



Insert 48 h(48) = 48 mod 13 = 9



Final Hash Table (Array form): [[195, 91], [], [], [16, 94, 81], [147], [265], [32, 162], [189, 202], [21], [48], [75], [180, 37], [207, 77]]

b) What is the maximum number of collisions caused by the above insertions?

7 Collisions in total (202, 91, 94, 162, 37, 77, 81)

Question 4
Assume an open addressing hash table implementation, where the size of the array N = 19, and the double hashing is performed for collision handling. The second hash function is defined as:
d(k) = q - k mod q, where k is the key being inserted in the table and the prime number
q = 11. Use simple modular operation (k mod N) for the first hash function.
a) Show the content of the table after performing the following operations, in order:
put(42), put(19), put(48), put(20), put(72), put(18), put(48),put(27), put(9).

put(42) Index = 42 mod 19 = 4

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 42 | | | | | | | | | | | | | | |

put(19) Index = 19 mod 19 = 0

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | | | | 42 | | | | | | | | | | | | | | |

put(48) Index = 48 mod 19 = 10

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | | | | 42 | | | | | | 48 | | | | | | | | |

put(20) Index = 20 mod 19 = 1

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 20 | | | 42 | | | | | | 48 | | | | | | | | |

put(72) Index = 72 mod 19 = 15

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 20 | | | 42 | | | | | | 48 | | | | | 72 | | | |

put(18) Index = 18 mod 19 = 18

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 20 | | | 42 | | | | | | 48 | | | | | 72 | | | 18 |

put(48) Index = 48 mod 19 = 10 [Collision]
d(48) = 11 – (48 mod 11) = 7 NewIndex = 10 + (1*7) = 17

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 20 | | | 42 | | | | | | 48 | | | | | 72 | | 48 | 18 |

put(27) Index = 27 mod 19 = 8

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 20 | | | 42 | | | | 27 | | 48 | | | | | 72 | | 48 | 18 |

put(27) Index = 9 mod 19 = 9

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 20 | | | 42 | | | | 27 | 9 | 48 | | | | | 72 | | 48 | 18 |

b) What is the size of the longest cluster caused by the above insertions?
Longest cluster size is 4

c) What is the number of occurred collisions as a result of the above operations?
1 collision occurred (Entering 48 for the second time)

d) What is the current value of the table's load factor?
9/19 = 0.4736842105

Question 5

Assume the utilization of linear probing instead of double hashing for the implementation given in Question 4. Still, the size of the array N = 19, and that simple modular operation (k mod N) is used for the hash function.

a) Show the contents of the table after performing the following operations, in order:

put(29), put(53), put(14), put(95), remove(53), remove(29), put(32), put(19), remove(14), put(30), put(12), put(72).

put(29) Index = 29 mod 19 = 10

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | 29 | | | | | | | | |

put(53) Index = 53 mod 19 = 15

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | 29 | | | | | 53 | | | |

put(14) Index = 14 mod 19 = 14

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | 29 | | | | 14 | 53 | | | |

put(95) Index = 95 mod 19 = 0

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | | | | | | | | | | 29 | | | | 14 | 53 | | | |

remove(53)

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | | | | | | | | | | 29 | | | | 14 | A | | | |

remove(29)

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | | | | | | | | | | A | | | | 14 | A | | | |

put(32) Index = 32 mod 19 = 13

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | | | | | | | | | | A | | | 32 | 14 | A | | | |

put(19) Index = 19 mod 19 = 0 [Collision #1] Therefore, index = 1 [Next available]

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | 19 | | | | | | | | | A | | | 32 | 14 | A | | | |

remove(14)

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | 19 | | | | | | | | | A | | | 32 | A | A | | | |

put(30) Index = 30 mod 19 = 11

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | 19 | | | | | | | | | A | 30 | | 32 | A | A | | | |

put(12) Index = 12 mod 19 = 12

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | 19 | | | | | | | | | A | 30 | 12 | 32 | A | A | | | |

put(72) Index = 72 mod 19 = 15

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 95 | 19 | | | | | | | | | A | 30 | 12 | 32 | A | 72 | | | |

Note:

For the remove method, a special character "A" is entered to denote that the cell has an available index to store a value. This is done so we do not encounter any issues when searching for a key in the hash map

b) What is the size of the longest cluster caused by the above insertions? Using Big-O notation, indicate the complexity of the above operations.

**Largest cluster size is 6**
**Best case – O(1), when there is no collisions occuring**
**Worst case – O(n).**
**When it is nearly full approximately the whole length of the hashmap would have to be checked linearly. The maximum positions checked would be n.**

c) What is the number of occurred collisions as a result of the above operations?

**1 collision (put(19))**