ASSIGNMENT 3

WRITTEN QUESTIONS PART

Data Structures and Algorithms

COMP 352

Laila Alhalabi # 40106558

Date: March 26, 2021

Question 1

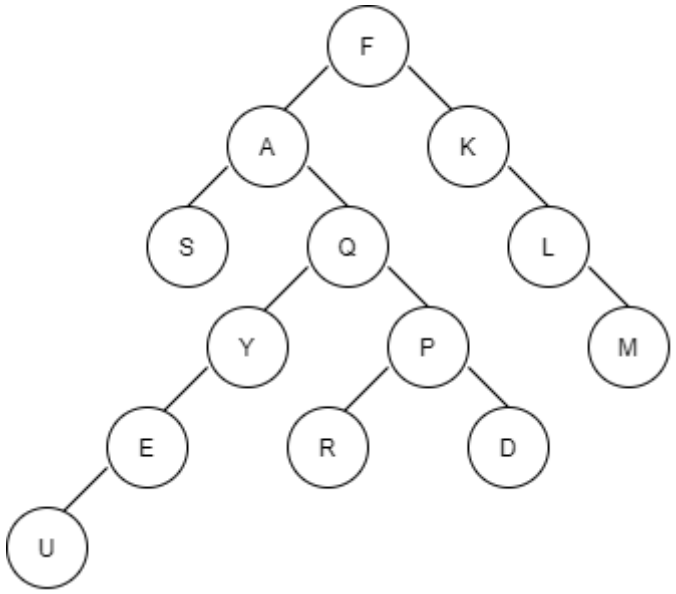
a) Draw a single binary tree that gave the following traversals:

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

a) Assume that the binary tree from the above- part (a)- is stored in an array-list as a complete binary tree as discussed in class. Specify the contents of such an array-list for this tree.

a)

The binary tree produced:



b)

- Index 0 starts from the base node F, left node's index is 2i+1 and right node's index is 2i+2.

- The parent index is (i-1)/2.

Contents will be (other indexes contents not mentioned in table are null):

index	0	1	2	3	4	5	9	10	11	19	21	22	39
value	F	A	K	S	Q	L	Y	P	M	E	R	D	U

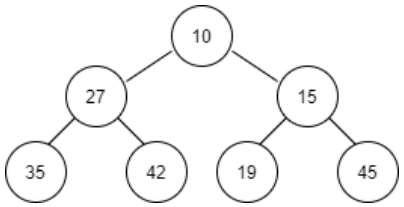
Question 2

Draw the min-heap that results from the bottom-up heap construction algorithm on the following list of values:

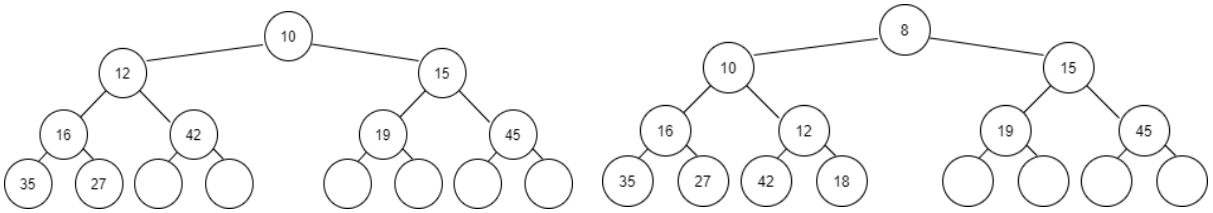
`10, 27, 15, 35, 42, 19, 45, 16, 12, 8, 18, 14, 13, 9, 20, 11, 13`

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.

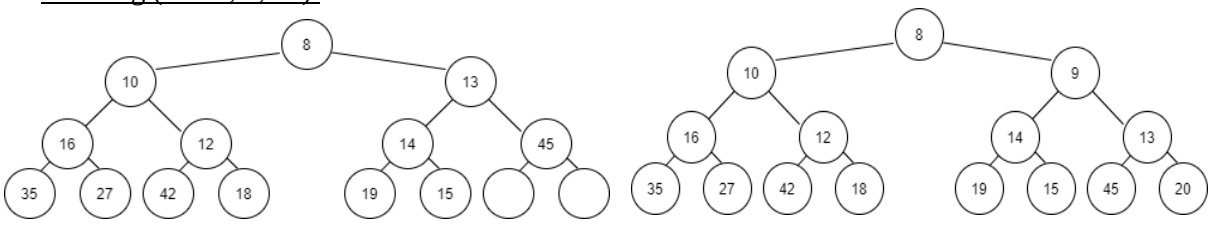
Inserting (10, 27, 15, 35, 42, 19, 45):



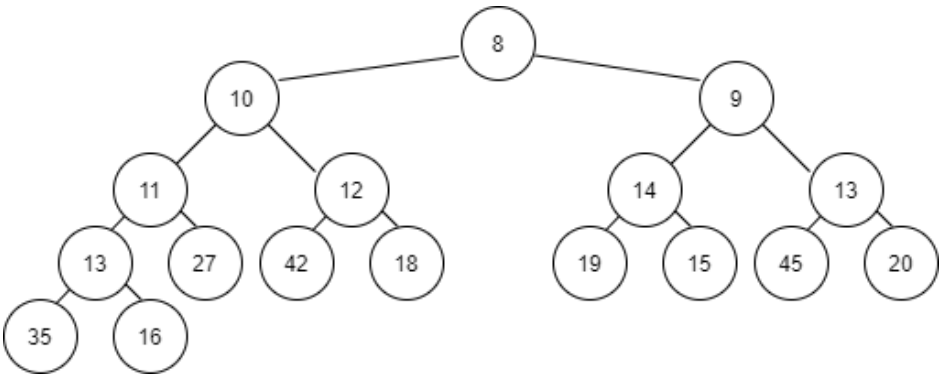
Inserting (16, 12, 8, 18):



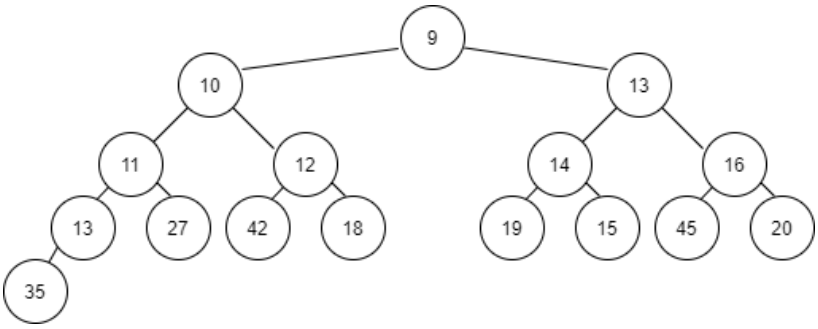
Inserting (14.13, 9, 20):



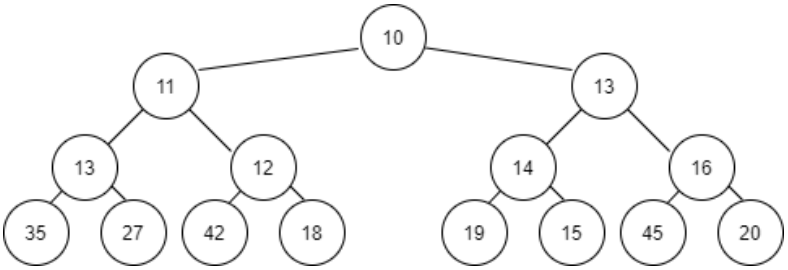
Inserting(11, 13), it becomes:



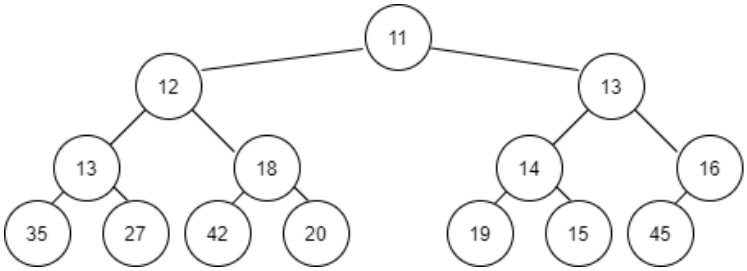
Removing first min value (8):



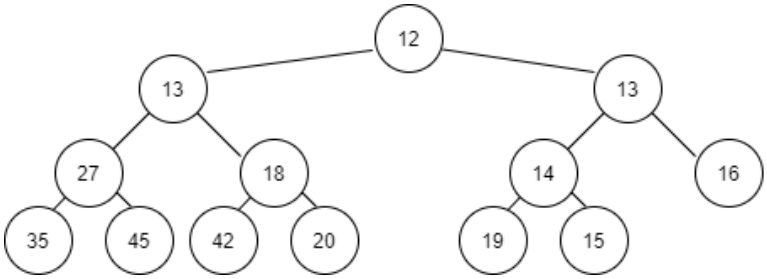
Removing first min value (9):



Removing first min value (10):



Removing first min value (11):



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: $h(k)=k \bmod 13$.

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}

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

a)

h(32) = 32 mod 13 = 6
h(147) = 147 mod 13 = 4
h(265) = 265 mod 13 = 5
h(195) = 195 mod 13 = 0
h(207) = 207 mod 13 = 12
h(180) = 180 mod 13 = 11
h(21) = 21 mod 13 = 8
h(16) = 16 mod 13 = 3
h(189) = 189 mod 13 = 7
h(202) = 202 mod 13 = 7
h(91) = 91 mod 13 = 0
h(94) = 94 mod 13 = 3
h(162) = 162 mod 13 = 6
h(75) = 75 mod 13 = 10
h(37) = 37 mod 13 = 11
h(77) = 77 mod 13 = 12
h(81) = 81 mod 13 = 3
h(48) = 48 mod 13 = 9

Index values calculated range is from 0 to 12.

Contents of the hash table without collision:

Index value	0	1	2	3	4	5	6	7	8	9	10	11	12
key	195	null	null	16	147	265	32	189	21	48	75	180	207

After collisions (no change/ collision indicated by '-'):

Index value	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
key	195	null	null	16	147	265	32	189	21	48	75	180	207
1st collision	91	-	-	94	-	-	162	202	-	-	-	37	77
2nd collision	-	-	-	81	-	-	-	-	-	-	-	-	-

b) Total collisions are **7** that happen at 6 indexes (0, 3, 6, 7, 11, 12).

Maximum number of collisions is **2** that happened at index **3**.

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).`

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

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

d) What is the current value of the table's *load factor*?

a)

put(42) = 42 mod 19 = 4

put(19) = 19 mod 19 = 0

put(48) = 48 mod 19 = 10

put(20) = 20 mod 19 = 1

put(72) = 72 mod 19 = 15

put(18) = 18 mod 19 = 18

put(48) = 48 mod 19 = 10

second hash: 11 – 48 mod 11 = 7

put(27) = 27 mod 19 = 8

put(9) = 9 mod 19 = 9

cluster	#1			#2		#3					#4		#5
Index value	0	1	2 to 3	4	5 to 6	7	8	9	10	11 to 14	15	16 to 17	18
key	19	20	null	42	null	48	27	9	48	null	72	null	18

b)

Total number of clusters is 5, and the size of longest one is 4

c)

The number of collisions is 1 (when k = 48).

d)

Load Factor = Keys / Size = 9/19 = 0.47

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).`

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.

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

a)

put(29) = 29 mod 19 = 10
put(53) = 53 mod 19 = 15
put(14) = 14 mod 19 = 14
put(95) = 95 mod 19 = 0
remove(53) = 53 mod 19 = 15 (53 deleted)
remove(29) = 29 mod 19 = 10 (29 deleted)
put(32) = 32 mod 19 = 13
put(19) = 19 mod 19 = 0 (inserted at key 1 instead of 0)
remove(14) = 14 mod 19 = 14 (14 deleted)
put(30) = 30 mod 19 = 11
put(12) = 12 mod 19 = 12
put(72) = 72 mod 19 = 15 (72 inserted at 15 since it got empty)

Cluster	#1			#2				#3	
Index value	0	1	2 to 10	11	12	13	14	15	16 to 18
key	95	19	null	30	12	32	null	72	null

b)

The longest cluster is #2, and its size is 3.

The worst complexity is O(*n*) when there is a collision, and best complexity is O(1) when there is no collision.

c)

Only one collision occurred when doing *put*(19) operation.