CS3340 Assignment 2

Daniel Rawana – 250911447

1.

After line 5:

A = {6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2}

B = { , , , , , , , , , , } (n = 11)

C = {2, 2, 2, 2, 1, 0, 2}

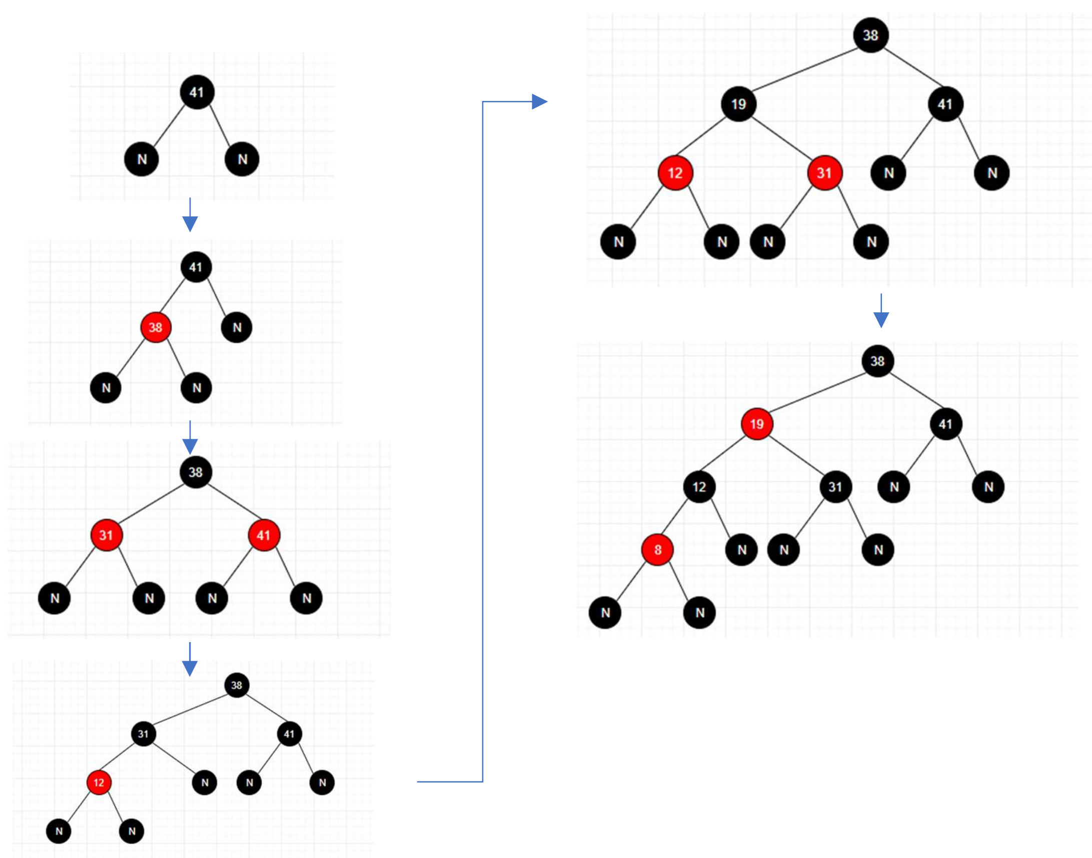
After line 8:

C = {2, 4, 6, 8, 9, 9, 11}

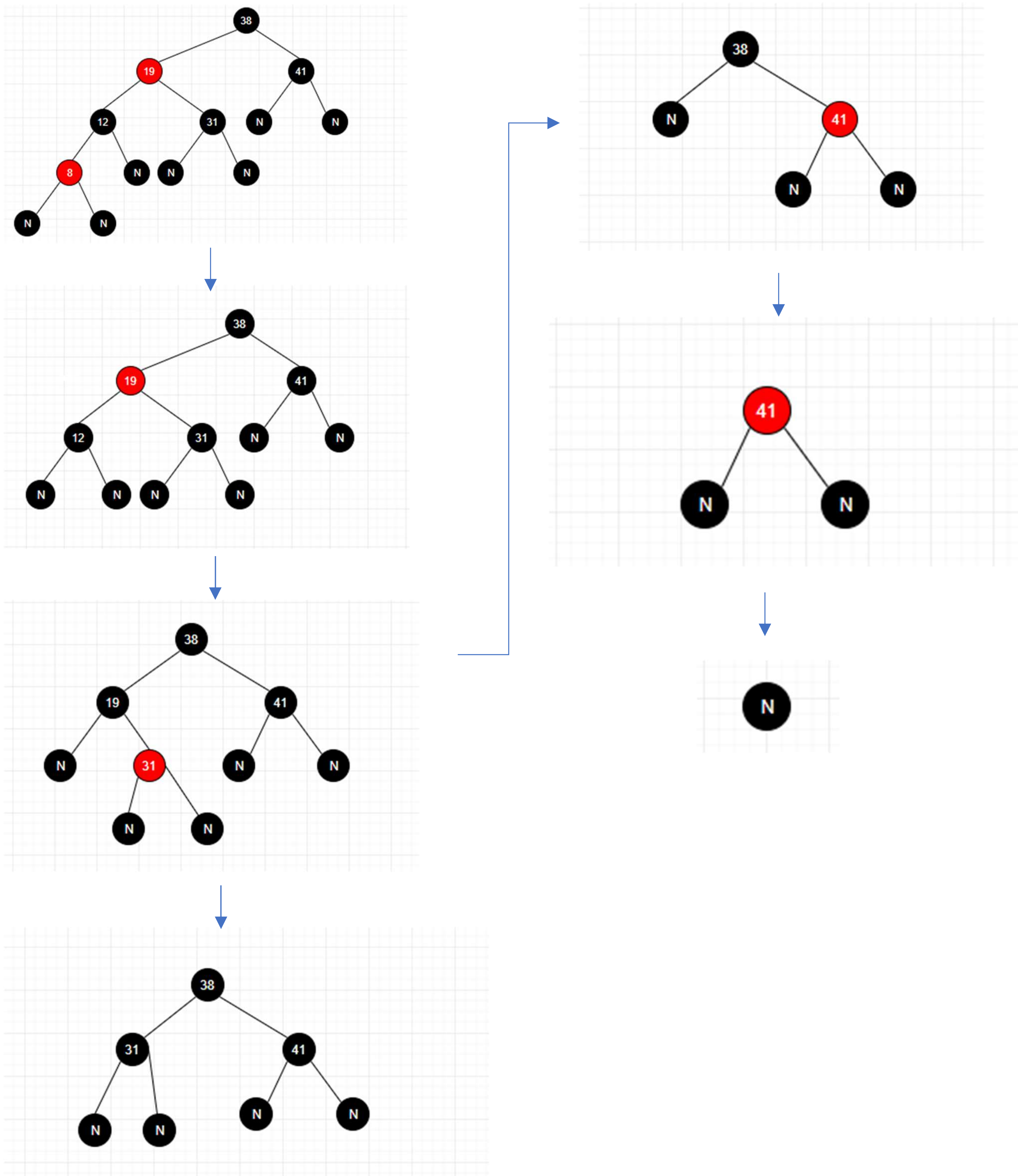
After one iteration of loop on line 10-12:

B = { , , , , , , 2, , , , } (n = 11)

2.



3.


38
19
41
12
31
N
N
8
N
N
N
N
N
N
38
19
41
12
31
N
N
N
N
N
N
38
19
41
N
31
N
N
N
N
38
31
41
N
N
N
N
38
N
41
N
N
41
N
N
N

4. We will use a divide-and-conquer approach similar to that of merge sort. We will continually merge and sort the k sequences in a pairwise fashion and will be left with $k/2n$ sequences where n is the number of merges performed. In the end we will be left with one sorted sequence.

This algorithm is correct because it acts the same as a merge sort except that it doesn't need to split the array.

Each iteration of the algorithm takes $O(nk)$ time since each merge requires 2n time with $k/2$ merges. There are $O(\lg k)$ iterations, so the overall time complexity is $O(nk \lg k)$.

5. A full binary tree that stores C will have exactly $2n – 1$ nodes. Perform an pre-order traversal of the tree in which each node is labelled with 0 if it an internal node and 1 if it is a leaf. Now we know that any character of C can be encoded in $\lg(n)$ bits, and therefore n characters can be encoded using $n[\lg n]$ bits.

6. Every node's rank starts at $0 = \log(1)$. Suppose the claim rank = floor(log(n)) also hold true for 1, 2, . . . , n nodes. The only operation that can change rank is UNION. Given n + 1 nodes, if we perform a UNION operation on two disjoint sets with *a* and *b* nodes where *a, b <= n.* The root of the first set has at most $\lfloor \lg a \rfloor$ and the root of the second set has at most $\lfloor \lg b \rfloor$. if the two ranks are unequal then the ranks are unchanged, but if they are equal the rank of the union increases by 1 and the new set's rank is $\lfloor \lg(n + 1) \rfloor$.

7. Since x.rank's value is at most $\lfloor \lg(n) \rfloor$, it can be represented using $\Theta(\lg(\lg(n)))$ bits.