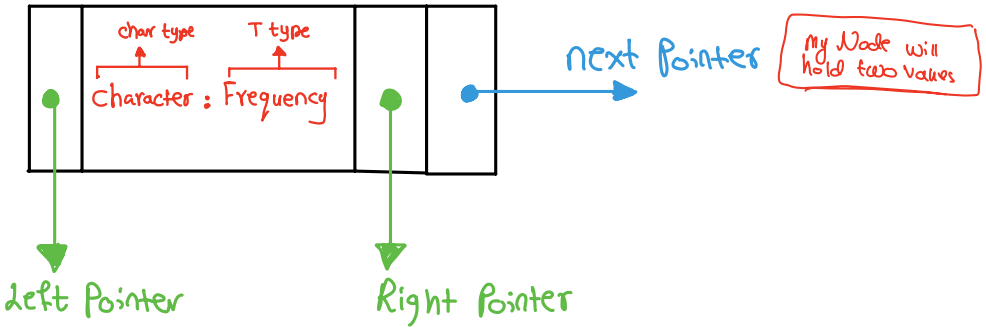
Diagrams to show how some of my methods Work.

① Node class:

How My Node look like:



② Linked List Class

1- I have two pointers here → head. / → tail, to make AddtoFirst, Add to End →

2- I have Attribute called size to track insertion and deleting from the Linked L

3- addcharacter method : First I will make a newNode !

this method is used to add the characters to a Sorted Linkedlist based on their frequency.
There is several scenarios that I covered in the method

① if the linked list was empty (which is head == null) the newNode will be the first node in my Linked List so make it head, tail.

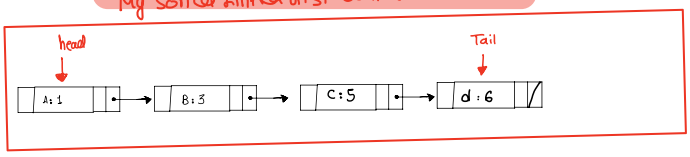
② second scenario: If the frequency of newNode is less than the head connect it to head, then make it new head!

③ Third scenario: If the frequency of newNode is greater than tail, so I will insert it to the end of the Linked list + MAKE it new tail!

④ Forth secenario, when non-of above happen I will iterate through my Linked list to find the correct position to my Node based on my frequency.

At the end increment the size! By one → size++;

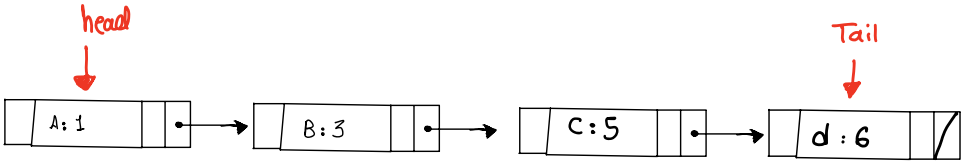
My Sorted Linked list will look like this



③ HuffmanTree Class:

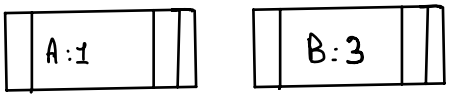
Constructing my Huffman Tree

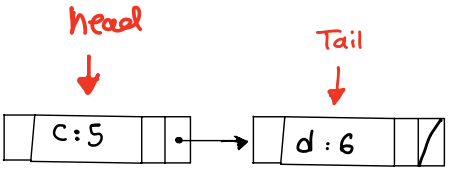
1- Build Huffman Tree : From a Sorted Linked List



- the constructing of the Huffman Tree goes as following: I will do this until the size of my Linked List is ONE

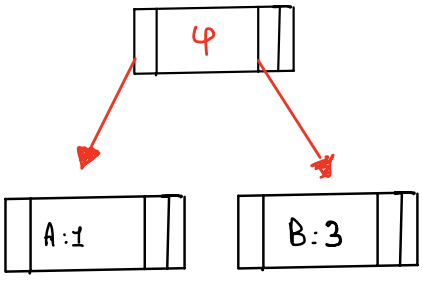
getValueOfFirst : this method in Linked List Class.
it will Always get the first node of my Linked List and return it
(means I will remove it and update head)!

① 



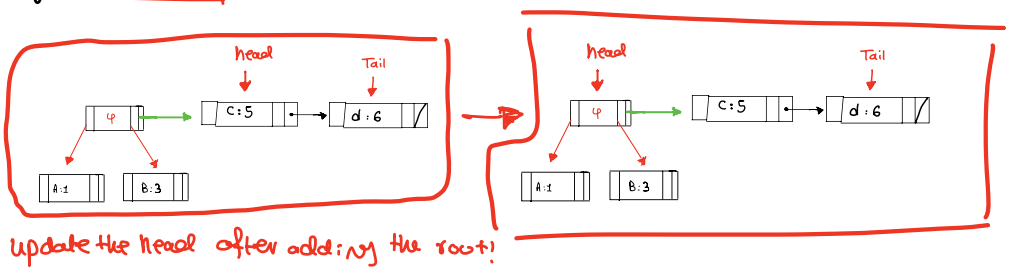
my Linked List will look like this. after removing first two nodes

② Add their frequency and make it the value of the root (make a new Node its value is the sum of their frequency).
(their parent)! , set the parent left and right to the Children!



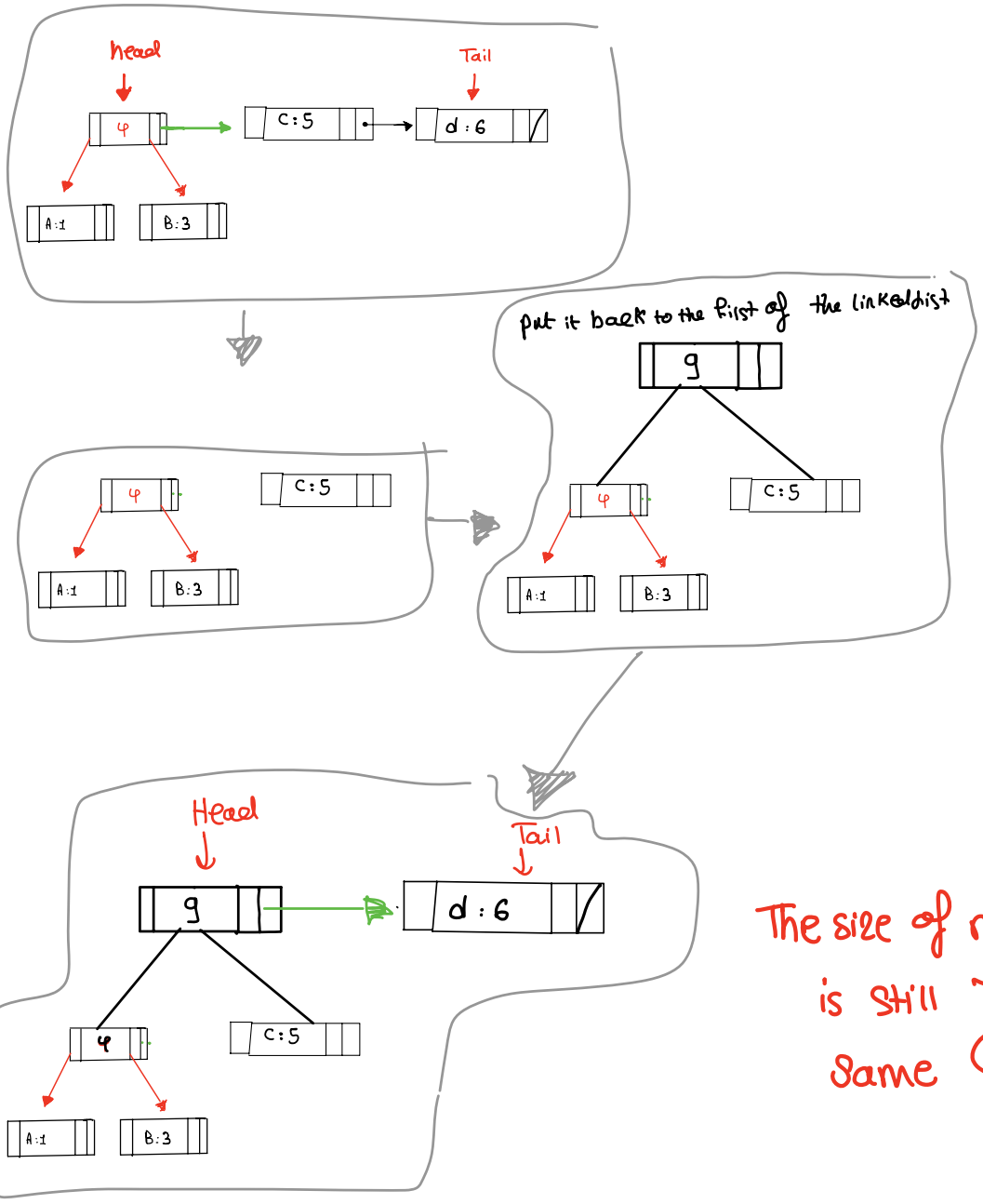
③ Add the root to the first of the linked list make it new head!

my Linked list

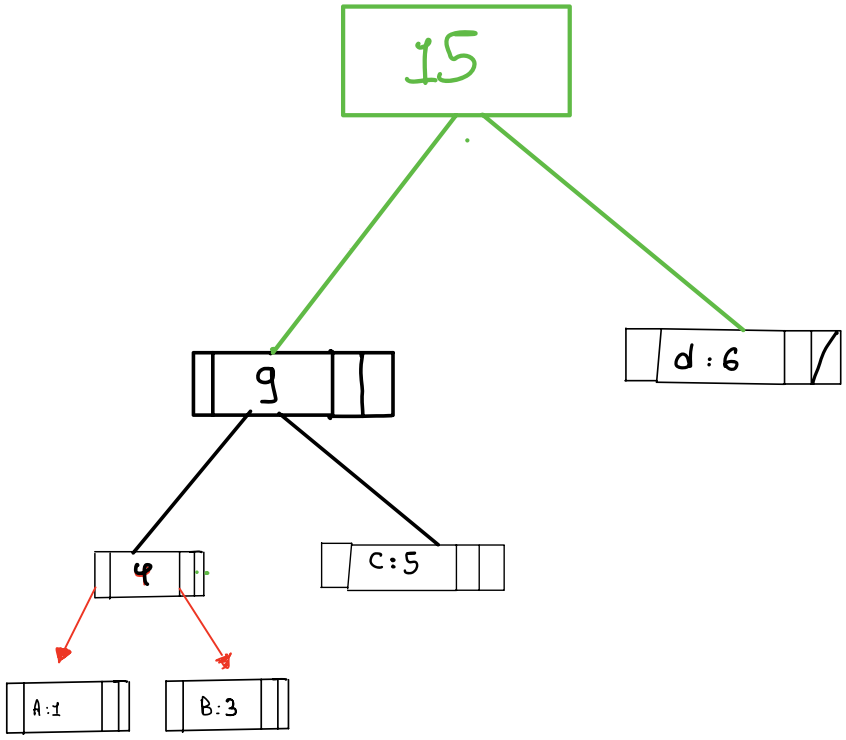


update the head after adding the root!

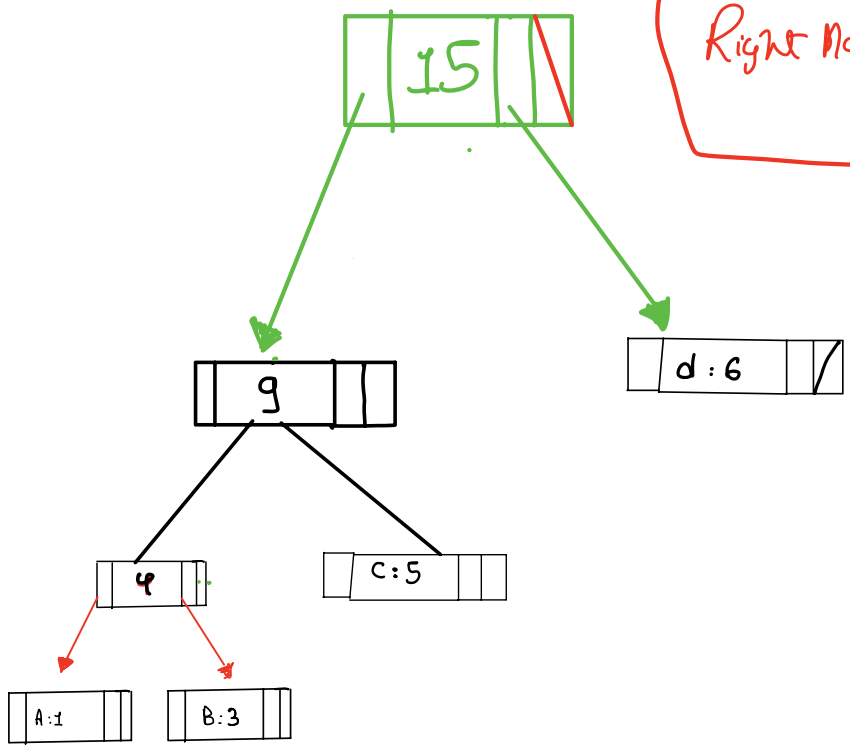
I will do these steps recursevly until my size of the linked list is one so I will make the only node root of all nodes!



The size of my Linked list is still >1 so Complete same Operation



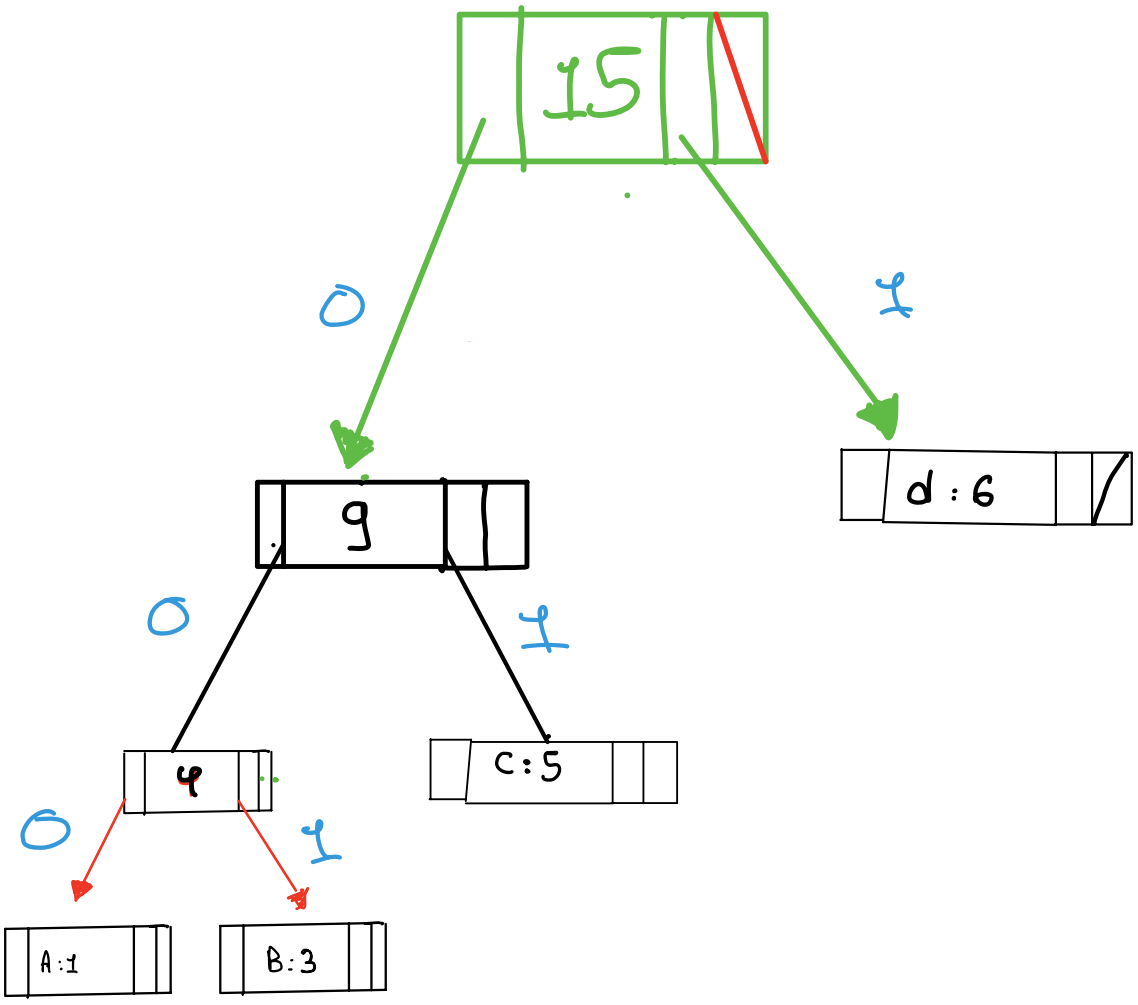
Add it to the Linked List!



Right now my size is not Greater than one !!

I will go out of the While loop! And 15 will be the Parent to all Nodes!

Encoding: Make every left Zero, every right 1!



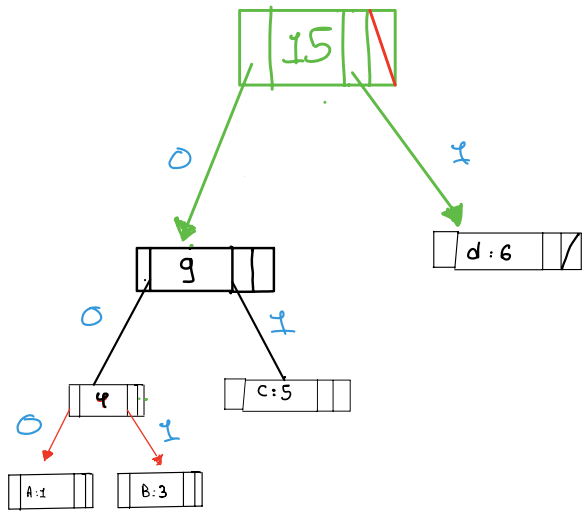
Encoding the characters: We will start from the root!

- A : 000 → less frequency more bit
  B : 001
  C : 01
  d : 1 → More frequency less bit

Decoding is the Opposite!

000 001 011

ALWAYS Start from the root!



000 001 011

A B C d