Question 1 :

C) If there is a constraint to 8 look ups, we can insert 4 bits in each node instead of 1.

First we convert the ip address to binary.

Once we have the ip address in binary, we can separate it in 8 blocks of 4 bits.

We know that there are 2^4 possible combinations of 0’s and 1’s in a string of 4 bits long.

For each one of them, we will assign a number as follows:

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Each node on the tree will contain 16 pointers.

Let’s give an exemple :

Lets say that the address to be added to the tree is : 190.152.20.19/32 A

Converted in binary : 10111110100110000001010000010011A

Now lets break this string into 8 block of 4 bits:

1011.1110.1001.1000.0001.0100.0001.0011

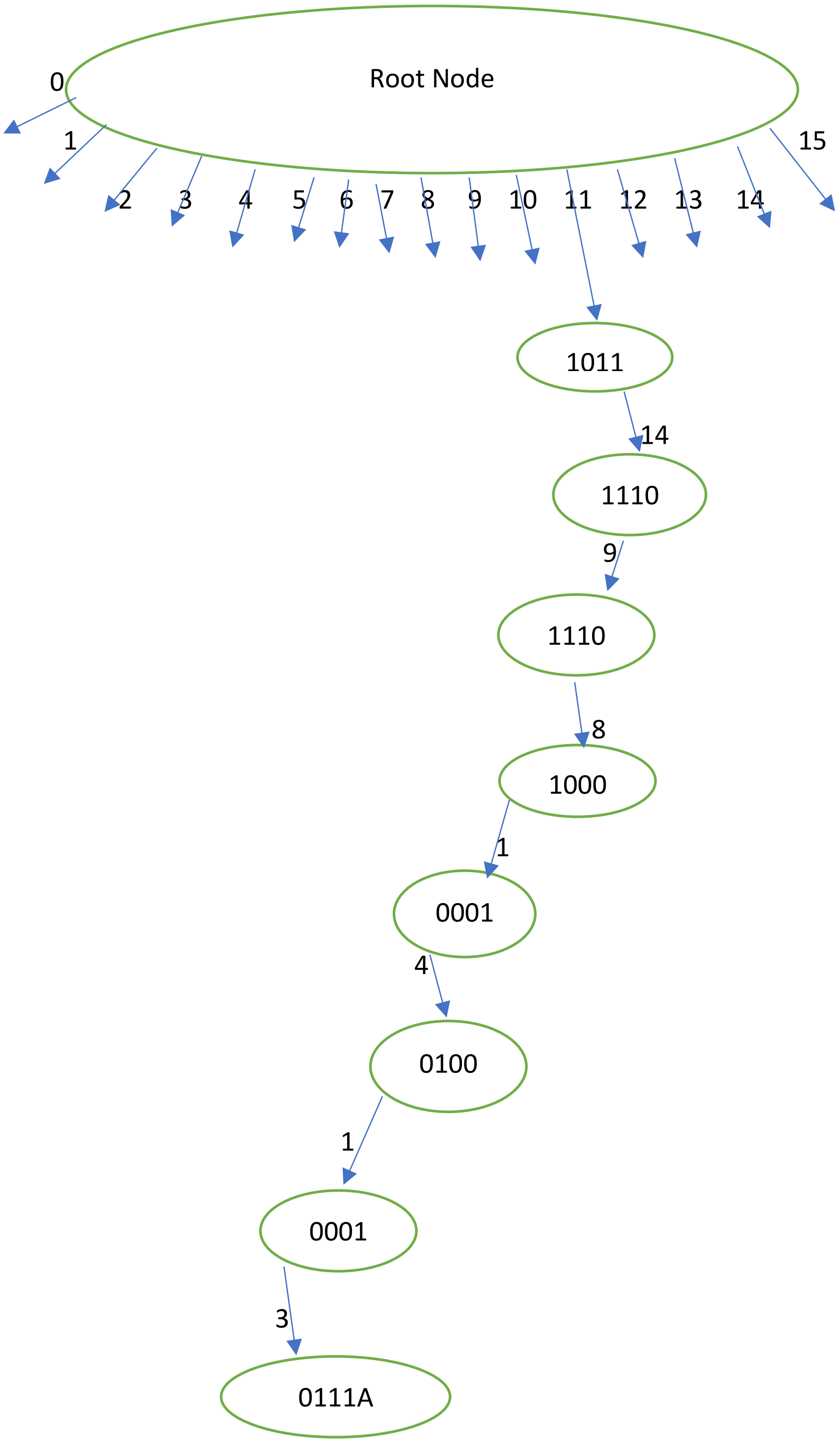
To add the string to the tree, lets run on each block.

The first block to add is 1011, so lets check up in the table above. 1011 corresponds to the number 11, which means that the string 1011 will be added to the 11th pointer of the root.

Now we want to add the next string 1110 , so in the table we see that it is corresponding to the number 14. So now, the 14th pointer of the node that we added previously will point to a node that contains the string 1110 and we continue so on.

(Please go to the next page)

Illustration :



Although I didn’t draw it, each node has 16 pointers.

We can see that the maximum look up is 8.

In the case where the prefix in not 32 bits like in the example, we should proceed as follows :

We can parse the string in blocks of 4 bits, if the prefix is an even number we have two cases:

1- 4 bits block is enough .
2- We can get an additional block with 2 bits only (but we never pass 8 blocks).

If the prefix is odd :

1-We can have an additional node that contains a string of one bit length

2-We can have an additional node that contains a string of 3 bits length

The last node will contain the action.