1. The figure below shows a data structure for IP address lookup using a trie. In the figure, the nodes containing numbers correspond to valid address prefixes and the numbers are the output ports that should be used by packets for which a given prefix is the best match. If a packet with destination address a3b2ff75 is received, what output port should it be forwarded on (the address is given in hexadecimal notation)?

What about packets with addresses 32fecd91 and 725cd401?

Both will go to 15.
Show how the data structure must be changed to include a new prefix 101101* with next hop 22. Show how it must be changed to include a new prefix 01001* with next hop 17.

Suppose the same prefix set were represented using the tree bitmap data structure described on page 1-27 of the notes, with a stride of 3. How many nodes does the data structure have? What is the depth of the deepest node (the depth of a node is the number of edges on the path from the root to that node)? What are the values of the internal and external bitmaps of the root node?

There will be 10 nodes in the data structure corresponding to the original trie. The depth of the deepest node is 2. The internal bitmap of the root node is 1100010 and the value of the external bitmap is 11111110.
2. The figure shown below shows an IP address lookup data structure based on a multi-bit trie with a stride of 3. Draw a picture of an equivalent binary trie data structure.