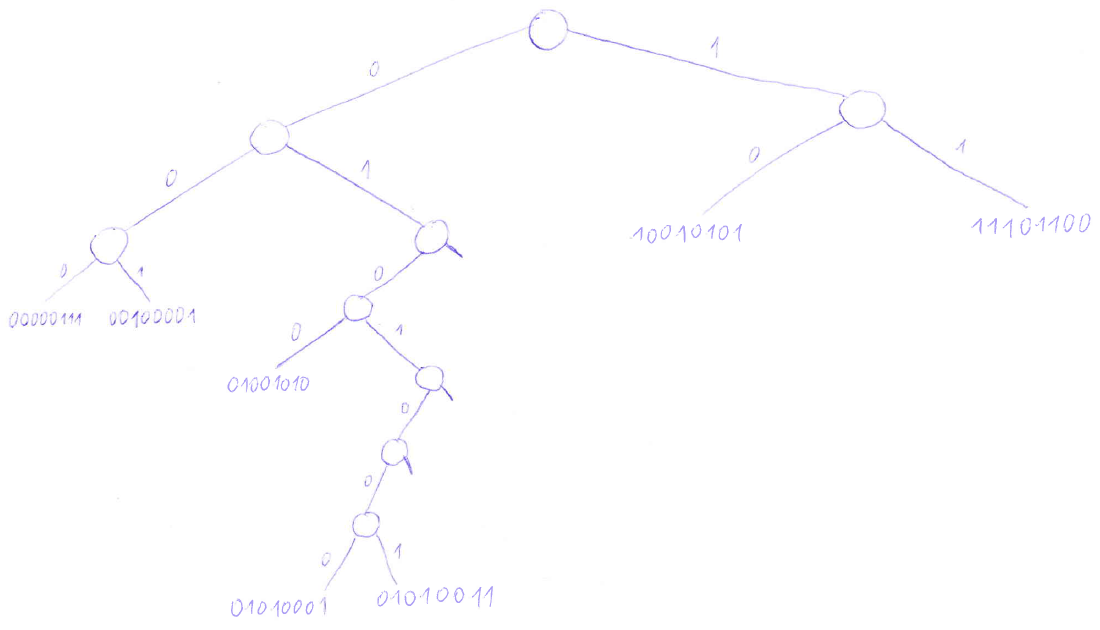
1. Build a binary trie by inserting keys in the following order

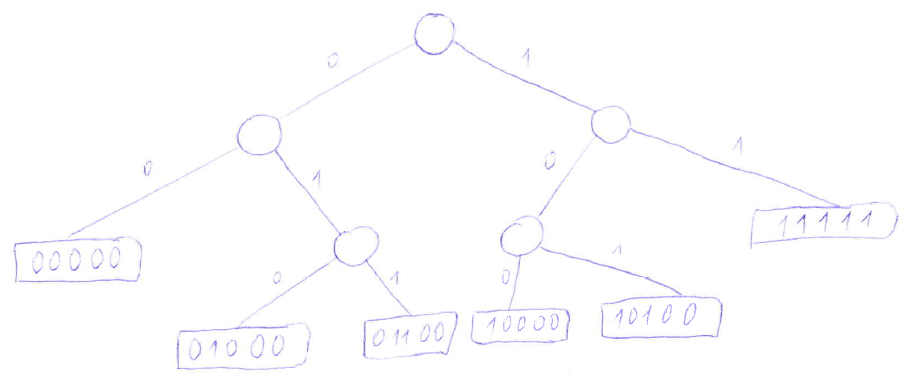
01010011, 00000111, 00100001, 01010001, 11101100, 00100001, 10010101, 01001010.

**Solution:**



2. Draw a trie which contains 6 keys, each of them of length 5 bits. Two of the keys must be 00000 and 11111. In addition, the trie depth is minimum possible.

**Solution:**



3. Determine the minimum and maximum memory needed to store a binary trie containg 1 000 000 binary strings, each of them consisting of 50 ASCII characters.

**Solution:**

Initial assumptions: Each inner node has two pointers, so we need 8+8=16 bytes to store it. Each leaf which contains a key requires 50 bytes. There are 1 000 000 leaves requiring 50 000 000 bytes. The remaining task is to count the minimum and maximum possible number of inner nodes.

The number of inner nodes is minimal if and only if there is no inner node whose left or right child is null. If we have k such inner nodes, there is room for k+1 leaves. Hence, we need 999 999 inner nodes and the total (minimum) memory is thus 50 000 000 + 16\*999 999 bytes.

Trie with the maximum number of inner nodes is constructed as follows: Take a balanced binary tree T having 500 000 leaves (the tree has 500 000 + 499 999 nodes). The height of the tree is =19. To create the trie, append to each leaf of T a chain of inner nodes of length 50-19-1=30 and to the last node of the chain append two trie leaves (containing keys that differ only in the last bit). The total number of inner leaves is 500 000 + 499 999 + 500 000\*30=15 999 999. This determines the total amount of the required memory.

Note that the calculation above is slightly imprecise since not all leaves of T have the same depth (some of them have depth 19, some of them have depth 18). This means that some of the chains can be even prolonged by one more node. But this is a really small imprecision.

4. Given a binary trie, the task is to replace each key by its binary complement (negation). Propose an efficient algorithm, which performs this task without building a new trie as well as without deleting the original one. What is the time complexity of the proposed algorithm?

**Solution:**

Use the following algorithm. Process nodes of the trie recursively starting by the root. Let X be the current node. If X is an inner node, swap its children (including their subtrees), then process the children recursively in the same manner. If X is a leaf which stores a key, change the key to its binary complement. Assuming there are *n* keys in the trie and each key has *N* bits, then the time complexity of the proposed algorithm is *O(nN)*.

5. Given two binary trie T1 a T2 where each contains unique keys (i.e., there is no key *k* contained in both, T1 and T2). Propose an algorithm Merge(T1, T2) which merges T1 and T2 into one trie. It is required that the algorithm does not create new leaf nodes, neither it deletes the existing nodes in T1 and T2.

**Solution:**

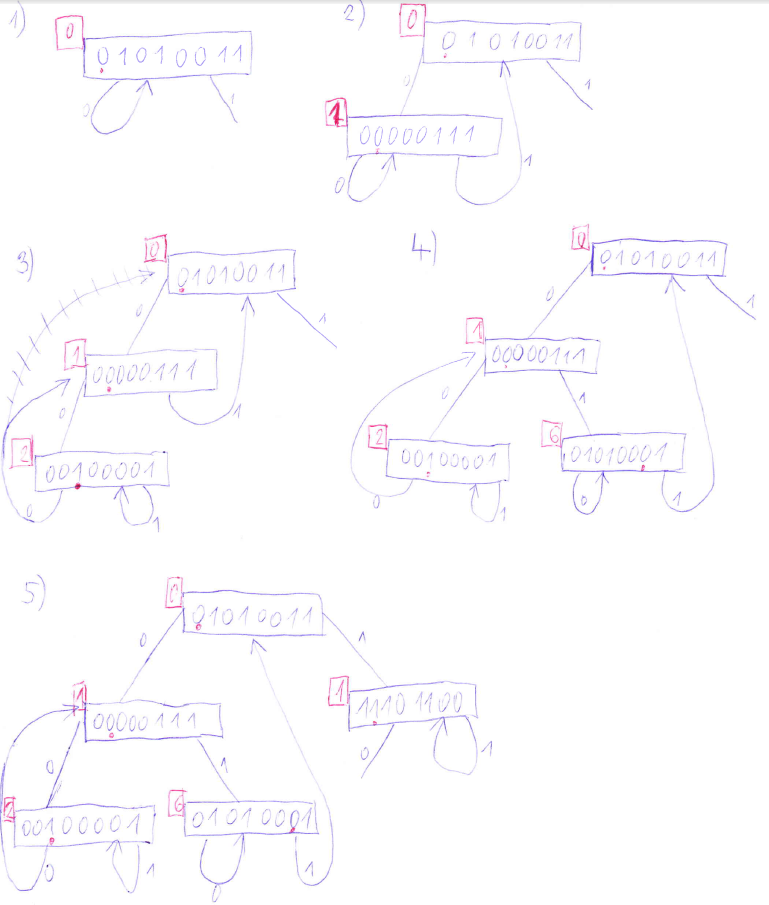
The algorithm performs the depth-first search simultaneously in both tries. It first visits the left child of a node, then it visits the right child. If this search finds a leaf with a key K1 in T1 and there is a subtree in T2 at the corresponding position, the algorithm moves the subtree from T2 to T1, then it inserts K1 in the moved subtree (this may result in creation of new inner nodes in T1). If the search finds a leaf with a key K2 in T2, it moves the node out of T2 and inserts it into the subtree in T1 at the corresponding position (again, it may result in creation of new inner nodes).

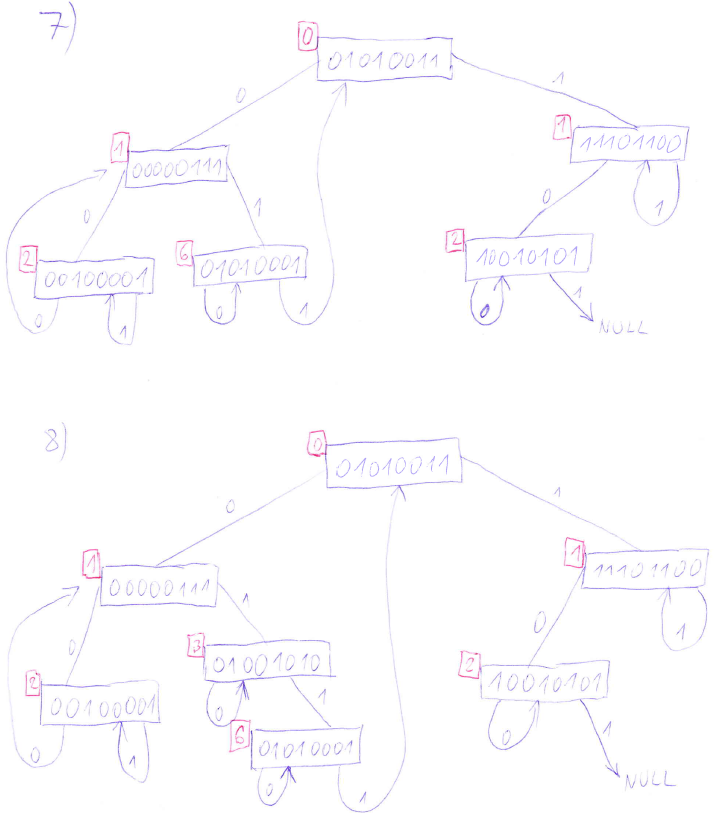
Solve the analogous assignments for patricia trie:

6. Build a patricia trie by inserting keys in the following order

01010011, 00000111, 00100001, 01010001, 11101100, 00100001, 10010101, 01001010.

**Solution:**

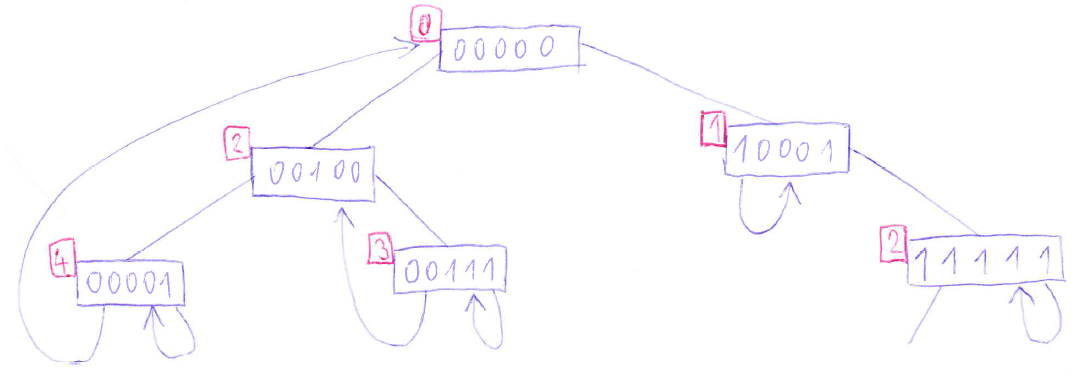




7. Draw a patricia trie which contains 6 keys, each of them of length 5 bits. Two of the keys must be 00000 and 11111. In addition, the trie depth is minimum possible.

**Solution:**

The following tree of height 2 is created by inserting keys (in the given order): 00000, 00100, 10001, 00111, 11111, 00001.



8. Determine the minimum and maximum memory needed to store a patricia trie containg 1 000 000 binary strings, each of them consisting of 50 ASCII characters.

**Solution:**

Each node of patricia trie stores a key and has two pointers. Moreover, the number of nodes equals the number of keys. This means that the minimum as well as the maximum memory needed to store patricia trie is 1 000 000 \* (16+50) = 66 000 000 bytes.

9. Given a patricia trie with binary keys, the task is to replace each key by its binary complement (negation). Propose an efficient algorithm, which performs this task without building a new trie as well as without deleting the original one. What is the time complexity of the proposed algorithm?

10 Given two patricia trie T1 a T2 where each contains unique keys (i.e., there is no key *k* contained in both, T1 and T2). Propose an algorithm Merge(T1, T2) which merges T1 and T2 into one trie. It is required that the algorithm does not create new nodes, neither it deletes the existing nodes in T1 and T2.