Date: Nov 19, 2025

Prefix Trees: 1-bit trie

- It is a tree like structure similar to a binary trie.
- Each node has 4 fields – left child, left data, right child and right data.
- Nodes at level l stores prefixes of length l. If the right-most bit in a prefix whose length is l is 0, the prefix is stored in the left data field of a node, i.e. at level l. Otherwise, the prefix is stored in the right data field of a node, i.e. at level l.
- At level i of a trie, branching is done by examining bit i (bits are numbered from left to right beginning with the number 1) of a prefix or destination address.
- When bit i is 0, we move into the left subtree and when it is 1, we move into the right subtree.
- The height of a 1-bit trie is order w, where w is the length of the longest prefix of the router table.

Disjoint sets and operations

- Disjoint sets are somewhat similar to mathematical sets but topics are changed so that they become useful to algorithms.
- The famous algorithm that uses disjoint sets is the Kruskal's algorithm which detects a cycle in a graph.

Suppose we have two components of a graph and their set representations are as follows:

<could not draw the graph>

The two sets represent two disjoint sets as they have no element in common.

Here we have two operations for disjoint sets:

- Find – Finds to which set a particular element belongs to.
- Union – We link vertex 4 to vertex 8, now 4 belongs to set S_1 and 8 belongs to set S_2. Since they are in two different sets, we find the union of two sets is S_3. If a new edge is created between 1 and 5, since 1 and 5 belong to the same set S_3, a cycle is formed.