BUILD BINARY TREE FROM ANCESTOR MATRICS

You are given the Ancestor matrix of a Binary tree, write an Algorithm to construct the corresponding tree.

For example, the below tree:

```
          10
         /  
        5   30
       / 
      4   8
     / 
    1   40
```

Will have the following ancestor Matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>4</th>
<th>5</th>
<th>8</th>
<th>10</th>
<th>30</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
```

The order of rows in the above matrix is not defined. I have kept it in the ascending order because the data in our nodes is numeric and can be ordered.
Essentially, in the ancestor matrix, each node has a row and a column (may not be the same). The value at $a[i][j]$ will be 1 iff node of Node representing $j$'th column is the ancestor of node representing the $i$'th row.

Write an algorithm that can construct the binary tree from a given Ancestor matrix.

**Note:** Since we don’t have information about whether a child is a left child or a right child, the tree which gets constructed will be unordered Binary tree (i.e, there can be max two children of a node but they will not be ordered as left or right).

**Solution:**

The row of the root node will have all zeros because there is no ancestor to the root node. We will use Queue to solve this problem. Let Q be the Queue with operations enqueue, dequeue and isEmpty

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>4</th>
<th>5</th>
<th>8</th>
<th>10</th>
<th>30</th>
<th>40</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

1. Add 1 more column in the matrix which contain the sum of all the elements in that row:

2. Find the row, which has all zeros (for which sum[i] = 0). lets call this node $r$ (root of the tree)
3. Q.enqueue ( r);
4. while (!Q.isempty)
   1. temp = Q.dequeue();
   2. remove both row & column of the temp node and update the sum column accordingly (ideally all the elements in the Sum column should decrease).
   3. Look for all the rows for which Sum[i] == 0
   4. add them as children to node temp.
   5. Insert them at the end of the queue.

The Algorithm will proceed as shown in the below diagram:

Now we have the root node (10). all the Nodes whose sum are zero will be children of this node. Hence the tree will look like:

```
10
/ \
5  30
```

10 will be dequeued from the queue and 5 & 30 are also inserted in the Queue.
Next element in the Queue (to be removed from Queue) will be 5. Remove the corresponding rows & columns and is sum value becomes zero corresponding to some nodes then insert them as child nodes of 5.

```
10
/ \
5  30
/ \
4  8
```

Also insert 4 & 8 in the Queue (and remove them from Matrix).

Repeat the same for 30. Then repeat the same for 4, 8 and other nodes.

Let me know if the above algo is not correct and you want the code for the same.

Source: http://www.ritambhara.in/build-binary-tree-from-ancestor-matrics/