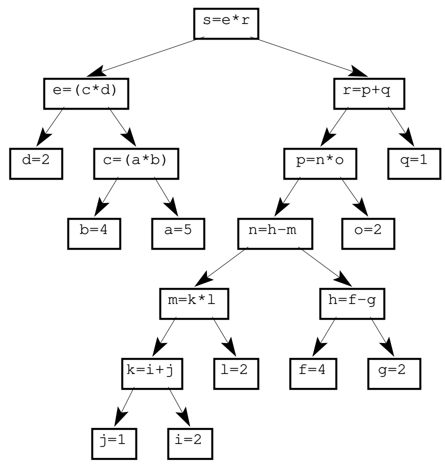
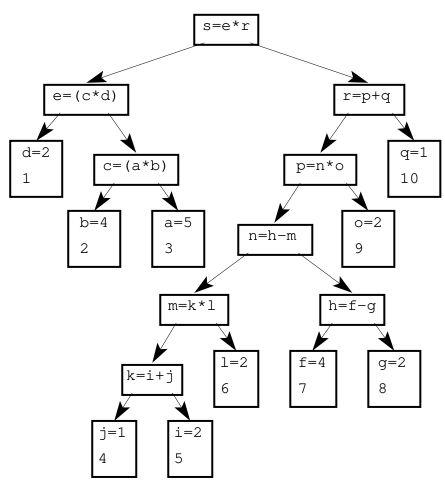
Jeff Hungerford

1.

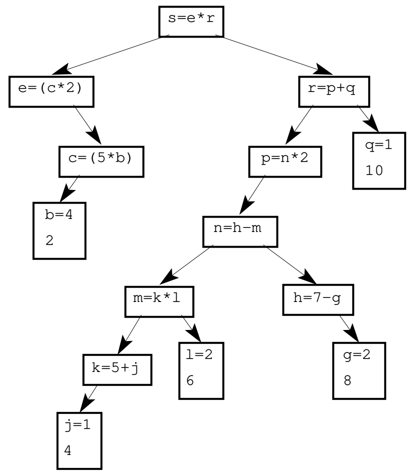
Start:



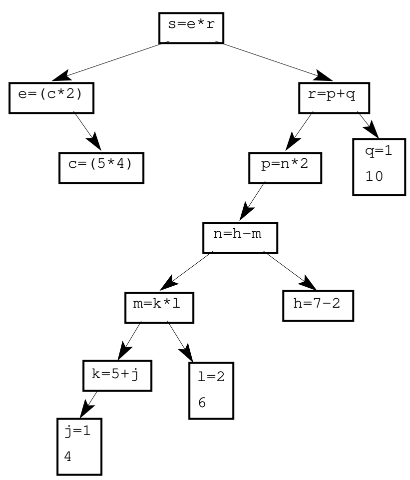
Number Leaf Nodes:



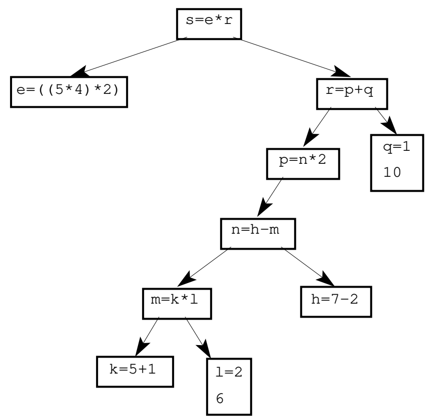
After Rake:



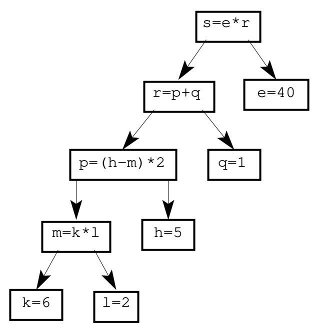
Left Compress:



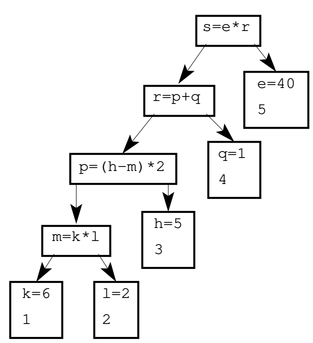
Compress



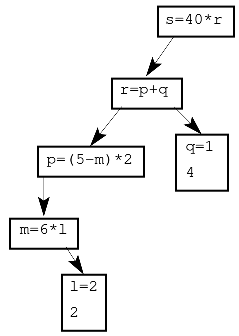
Simplify:



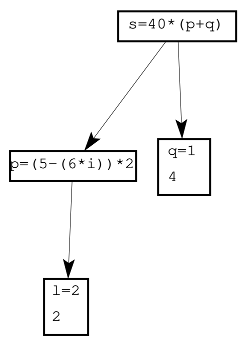
Number:



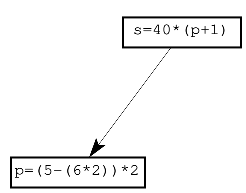
Rake:



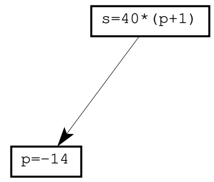
Compress Left:



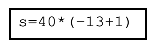
Compress Right:



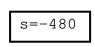
Simplify



Compress:



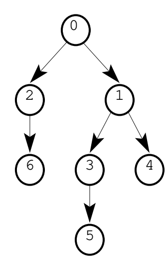
Simplify:



2.

3.

From a list of node traversals resulting from an Eulerian tour, the decendant of a node of a tree can be determined by the following procedure:



For a node X, the number of decendant is the position in the tour of the traversal from X to it's parent, minus 1, minus the position of the traversal from the parent of X to x, divided by two.

For the above tree, the serves of traversals making up the Eulerian tour are:

$$\left\{\begin{array}{l}(0 \rightarrow 2), (2 \rightarrow 6), (6 \rightarrow 2), (2 \rightarrow 0) \\ (0 \rightarrow 1), (1 \rightarrow 3), (3 \rightarrow 5), (5 \rightarrow 3) \\ (3 \rightarrow 1), (1 \rightarrow 4), (4 \rightarrow 1), (1 \rightarrow 0)\end{array}\right\}$$

To demonstrate, I will compute the number of decendants of node 3, and the number of decendants for node 1.

The parent of node 3 is node 1.

The location of the traversal from node 1 to node 3 $\{(1 \to 3)\}$ in the tour is 6.

The location of the traversal from node 3 to node 1 $\{(3 \to 1)\}$ in the tour is 9.

$$\frac{(9-1-6)}{2} = 1$$ so node 3 has one decendant

The parent of node 1 is node 0.

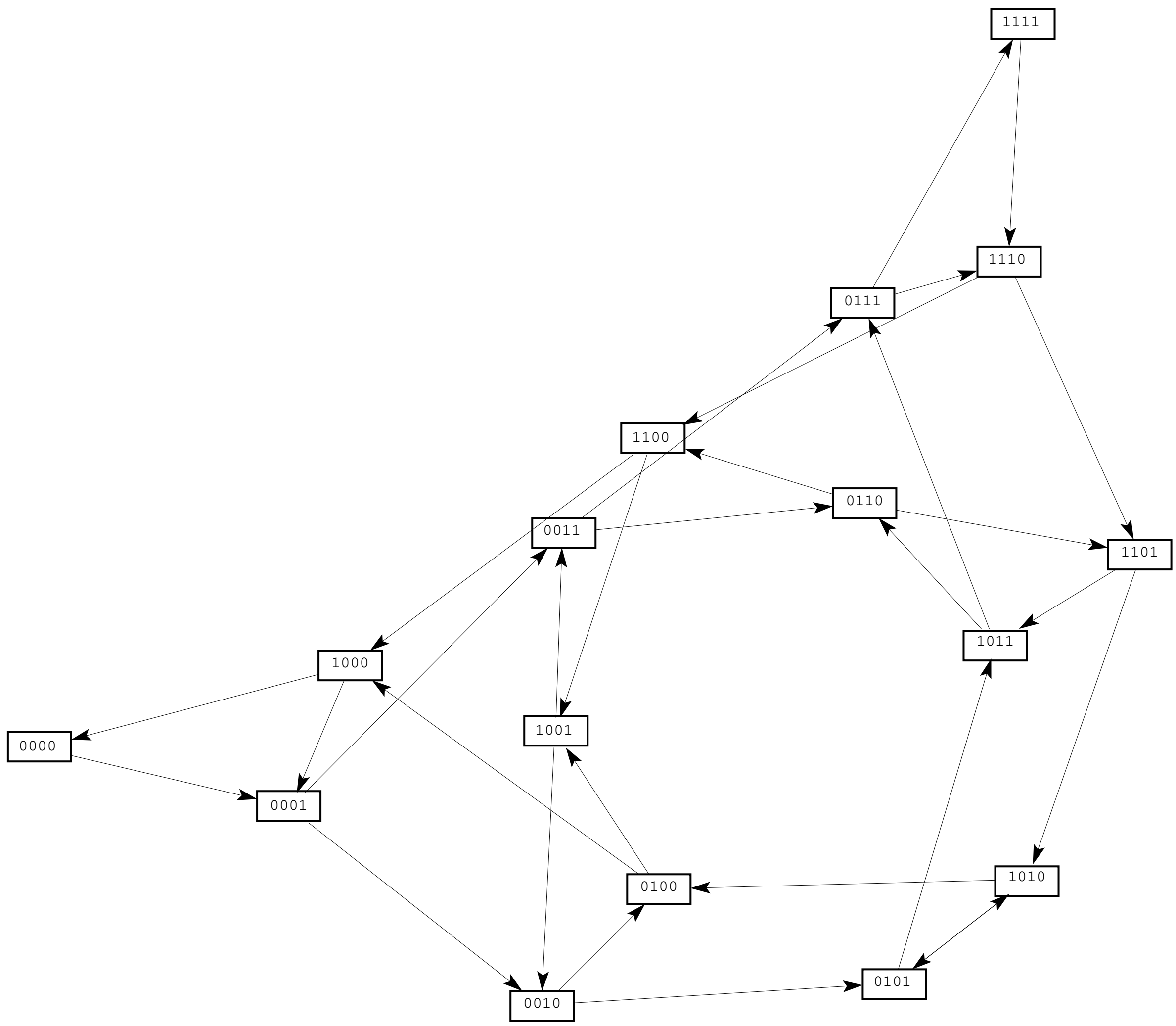
The location of the traversal from node 0 to node 1 $\{(0 \to 1)\}$ in the tour is 5.

The location of the traversal from node 1 to node 0 $\{(1 \to 0)\}$ in the tour is 12.

$$\frac{(12-1-5)}{2} = 3$$ so node 1 has three decendants.

4.

The following figure is a 16 node (4 bit language) De Brujin's graph.



5.

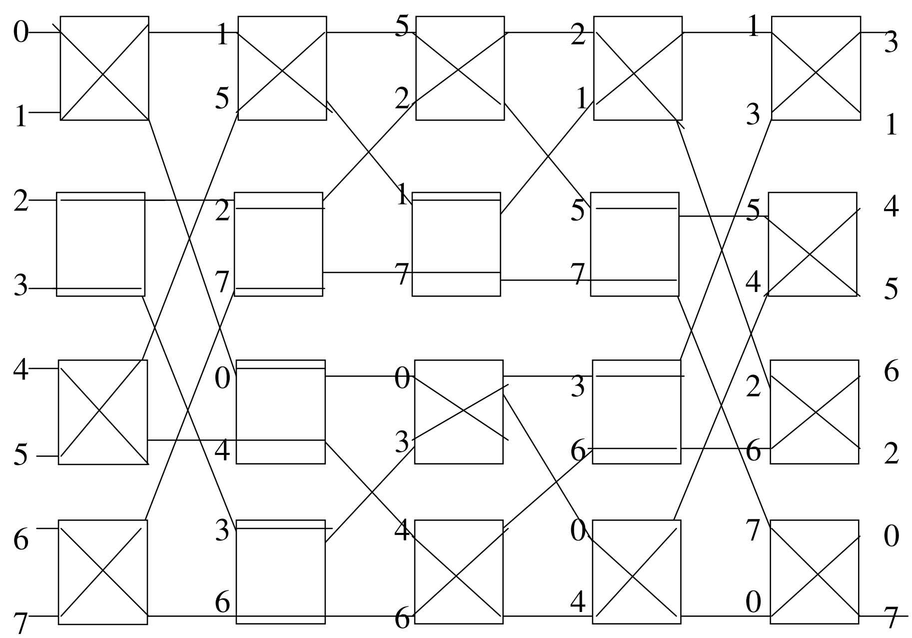
6.

7.

8.

A)

The following Benes network preforms the desired permutation.



B) Yes, as the bernes network is capable of preforming an arbratary permutation, if the switches are replaced with comparitors, it is capable of preforming a sort.

C) Yes, If a network si cabable of sorting an arbratary input sequence, it is capable of preforming any permutation on it's input values. By locking the decisions of the comparators, and turning them into switches, we can convert a sorting network into a network which preforms a permutation.