## Problem

Show that the parity function with n inputs can be computed by a branching program that has O(n) nodes.

## Step-by-step solution

**Step 1** of 1

**A branching program** is defined as “**a directed acyclic graph** where labels of all the nodes are maintained by the variables, except for two output nodes labeled 1 or 0. Here, all the nodes **whose labels are maintained by the variables** are called **query nodes**. Every query nodes consists of two outgoing edges: one is labeled 1 and another one is labeled 0.

- So, from the definition of branching program as defined above “the n-input parity function can be computed by a branching program that consists $O(n)$ nodes.
- It can be achieved by **building a binary tree of gates that compute XOR function**, **where XOR function is used as equivalent to the parity function. The implementation of each XOR gate can be done by using two AND’s, two NOT’s and one OR gates.**
- As the implementation of XOR function consists a binary tree of different gates. Here, the output node of binary tree is labeled 1 and 0, which acts a branching program.

**Therefore**, from the above explanation it can be said that “**the n-input parity function can be computed by a branching program that consists** $O(n)$ **nodes**”.

Comment