Implementation of Quantum Galton Board.

Introduction: The project presents a quantum version of the Galton Board (a device used to demonstrate probability and the normal distribution) and shows how to implement it on a quantum computer more efficiently. Although- there exist the classical approach in which a ball hits the peg and produces a 50% probability of going left or right, then hitting another peg on the board. After many drops, the ball forms a bell-shaped (normal) distribution at the bottom. In the quantum Galton board, instead of hitting (simulating) each ball path one by one, a quantum computer can put all possible paths into superposition and process them at once. This means exponential speed up of 2n^2 path can be calculated using only O (n^2) resources.

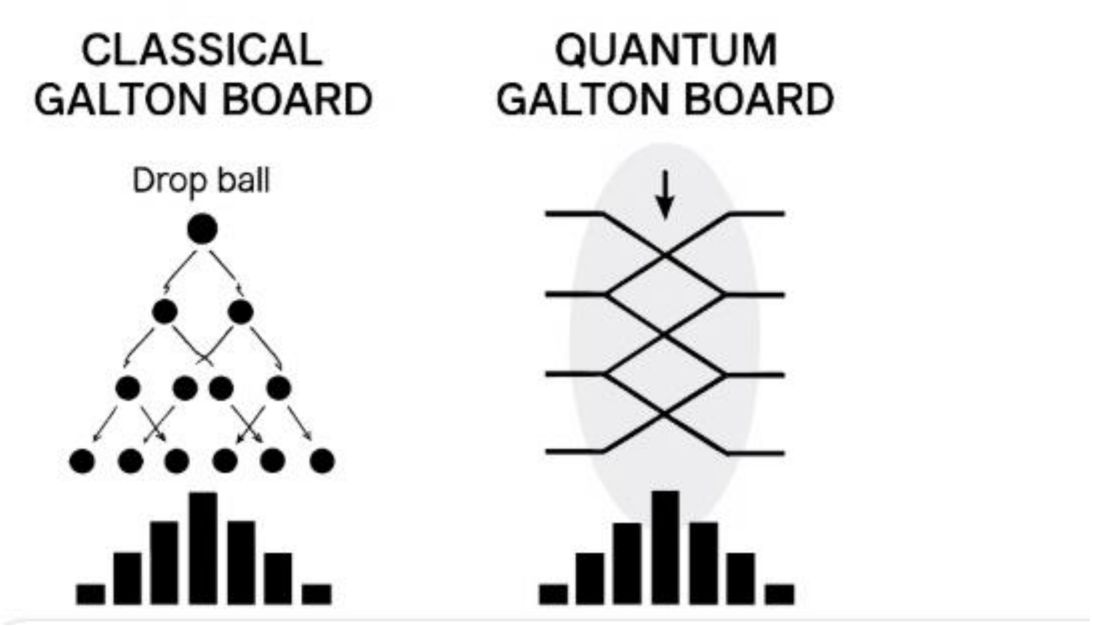

Figure1.0 is a classical and quantum Galton board.

In the diagram above:

The classical Galton board: A ball is dropped through rows of pages. Each time the ball hits the peg, it goes left or right with 50% probability. It then drops to form a bell-shaped distribution at the bottom.

The Quantum Galton Board: Here, instead of simulating each ball path one by one, all the path is put in superposition and process them at once by a quantum computer.

Flow of Quantum Galton Board

Conceptual Steps:

1. **Initialize Qubits**
   - All qubits start in ∣0⟩|0\rangle∣0⟩
   - Add **one "ball" qubit** set to ∣1⟩ with an X gate
2. **Add Control Qubit for Each Peg**
   - Apply **Hadamard** (or Rx(θ) for bias) to make 50–50 superposition of left/right paths
3. **Simulate Peg with Controlled-SWAP**
   - Controlled-SWAP moves the "ball" to left or right track depending on control qubit state
4. **Repeat for Multiple Levels**
   - Reset control qubit each time so it can be reused for the next peg row
5. **Measure**
   - Output qubits are measured to determine the final "landing position" of the ball(s)
   - For unbiased: normal distribution
   - For biased: skewed distribution

Below are the step by step plan to build in Qiskit.

Step 1: Environment setup

Step2: Implement 3 Quantum Peg

STEP3: Build multi- Peg QGB

STEP 4: Run on a simulator

STEP 5: Run on a real quantum hardware

STEP6:Post- Processing

STEP7: Extension