Grover’s Algorithm properties:

**Let:** applies Grover’s algorithm using oracle where , with Grover iterations.

returns a set of all input states that are phase flipped by oracle

returns a Grover’s oracle from ,

**Where:**

returns a Grover’s oracle ,

**Where:** The phase is flipped for all states in

Another property that I’m not sure how to test is accuracy and number of grover iterations,

* **Inputs**: where
* **Precondition**:
* **Operation**:
* **Postcondition**:
* **Inputs**: where
* **Precondition**:
* **Operation**:
* **Postcondition**:

Grover’s only works if Marks <= half of n, so we double n by adding a register

* **Inputs**: where
* **Precondition**:

**Operation**:

* **Postcondition**:
* **Inputs**: where
* **Precondition**:

**Operation**:

* **Postcondition**:
* **Inputs**: where
* **Precondition**:

**Operation**:

* **Postcondition**:

* **Inputs**: where
* **Precondition**:

**Operation**:

* **Postcondition**:

* **Inputs**: where
* **Precondition**:
* **Operation**:

* **Postcondition**:

1. remove a qubit from oracle, should still be able to identify output if N is big enough.
2. remove a qubit from oracle, half the number of marked states.