**Properties:**

1. When estimating the phase of a unitary gate, if QPE is ran with different amounts of estimation qubits, the QPE run with more estimation qubits will be closer to the expected value.
2. When estimating the phase of a unitary gate, if QPE is ran with different amounts of estimation qubits, the QPE run with less estimation qubits will be further to the expected value.
3. Most results generated by the algorithm should be for the qubit string that generates the closest phase to the one entered into the algorithm.
4. The same bitstring should be generated (100%) with certainty when phase entered is an exact multiple of where N = amount of estimation qubits and N > 0. (Assuming the simulator is without noise)

**Properties (new):**

1.

Precondition:

Qubit amount for estimation 1 : n

Qubit amount for estimation 2 : m

m > n

Randomly selected angle between 2pi and 0 : theta

(at least 10000 shots of results)

Operation:

QPE(n, theta, 10000) : (most frequent result) = estThetaN

QPE(m, theta, 10000) : (most frequent result) = estThetaM

Output:

assertTrue(abs(estThetaN-theta) > abs(estThetaM-theta)))

(in words, estimated theta with M qubits, should be closer to estimated theta with N qubits )

2.

Precondition:

Qubit amount for estimation : n

Randomly selected angle between 2pi and 0 : theta

(at least 10000 shots of results)

Operation:

QPE(n, theta, 10000) : (most frequent result) = estThetaN

Output:

assertTrue(abs(estThetaN-theta) <= 2^-n))

(estimated phase should be accurate to 2^-n radians)

3.

Precondition:

Qubit amount for estimation : n

Randomly selected angle between 2pi and 0 s.t. it’s also an exact multiple of

: theta

(at least 10000 shots of results)

Operation:

QPE(n, theta, 10000) : (most frequent result) = estThetaN

Output:

assertEqual(estThetaN, theta)

assertEqual(count(estThetaN), 10000)

(QPE should output the exact phase, if input rotation is a multiple of )

(There should only be one output with a count of 10000, which is the accurately estimated phase)

4.

Precondition:

Qubit amount for estimation : n

Randomly selected angle between 2pi and 0 : theta

Any shots : s

Operation:

QPE(n, theta, s) : (most frequent result) = estThetaN

Output:

assertTrue(estThetaN\*2\*pi >= 0)

assertTrue(estThetaN\*2\*pi <= 2\*pi)

Output answer should always be between 0 and 2pi radians