Exercise Set 4 (graded)
Quantum Computation

**Exercise 1** *Deutsch's algorithm*

This algorithm was first imagined by David Deutsch (and then extended together with Richard Josza, to become the algorithm that was presented in the lectures).

The problem is the following : consider a simple Boolean function $f : \{0,1\} \to \{0,1\}$; how many evaluations of this function $f$ are needed in order to decide whether $f(0) = f(1)$ or $f(0) \neq f(1)$? Classically, it is clear that exactly two evaluations of the function $f$, namely the evaluations of both $f(0)$ and $f(1)$, are needed in order to answer this questions. Deutsch imagined a quantum algorithm that could answer the question with a single evaluation of the function $f$, more precisely a single call to the quantum oracle $U_f$ with 2-qubits input and 2-qubits output, defined as follows:

$$U_f(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |y \oplus f(x)\rangle$$

(a) Construct the quantum circuit for the oracle $U_f$, for each of the 4 possible functions $f : \{0,1\} \to \{0,1\}$.

(b) Reconstruct then the complete quantum circuit seen in the lectures in this particular case and compute the output of the circuit, explaining how the final measurement allows to decide between the above two alternatives ($f(0) = f(1)$ or $f(0) \neq f(1)$).

*Notes:* - For part (b), *please do not copy-paste* the lectures with the same notations ; rather, redo the exercise from scratch in this particular case !

- The "quantum advantage" obtained here is mostly theoretical, as it assumes that the oracle $U_f$ is given to us. But building this oracle requires knowing the function $f$...

**Exercise 2** *Bernstein-Vazirani's algorithm*

Consider a vector $a = (a_1, \ldots, a_n) \in \{0,1\}^n$, as well as the function $f : \{0,1\}^n \to \{0,1\}$ defined as

$$f(x) = a \cdot x = a_1 x_1 + \cdots + a_n x_n, \quad \text{for } x \in \{0,1\}^n$$

Classically, $n$ evaluations of the function $f$ are needed in order to discover the value of vector $a \in \{0,1\}^n$, by considering successively $x^{(1)} = (1, 0, \ldots, 0)$, $x^{(2)} = (0, 1, 0, \ldots, 0)$, etc.

(a) Show that assuming again the existence of a quantum oracle $U_f$ with $n+1$-qubits input and output, as defined in the lecture on the Deutsch-Josza algorithm, it is possible to discover the value of the vector $a$ with probability 1 and a single call to the oracle $U_f$.

(b) Consider now a slightly different function $f$, defined as:

$$f(x) = b \oplus a \cdot x, \quad \text{for } x \in \{0,1\}^n$$

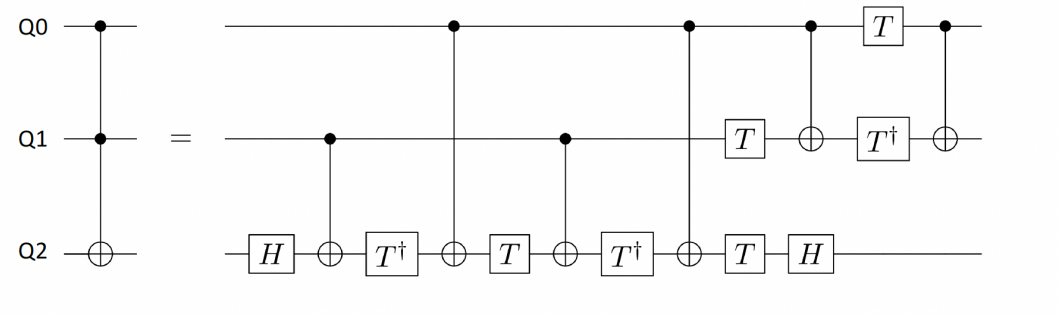
where $b \in \{0,1\}$ is also unknown a priori.

(b.i) With the same circuit as above, is it still possible to discover the value of the vector $a$ with probability 1 and a single call to the oracle $U_f$?

(b.ii) And what about the value of $b$? Is it possible to determine this one?

**Exercise 3** *IBM Q practice: Implementation and tests with the Toffoli gate*

In Homework 3, you proved the following circuit identity:




Note that with respect to Homework 3, we have removed the end gates T and S on the first two qubits. These qubits just add some phase factors with respect to the standard Toffoli gate which make no difference (at least for this exercise). You are asked to test this identity with Qiskit and IBM Q platform for multiple entries of your choice from the computational basis, for example $|000\rangle$, $|110\rangle$, $|111\rangle$. Concretely: produce and extract pdf pictures of histograms of measurements for *the circuit on the right hand-side of the identity* with 1024 shots.

(a) Run the circuit on an ideal simulator.

(b) Choose a real machine (tell which one). Transpile your circuit to match this device. What are the depths of the original circuit and the new one?

*Note:* Transpilation is a process of rewriting and optimizing a given circuit to match a quantum device. You can see an example in the tutorial.

(c) Obtain the noise model from the real machine (see the example from this link). Construct a simulator with this noise model by passing it as an argument `noise_model` when constructing `AerSimulator`. Run the circuit on this noisy simulator.

(d) Run the circuit on the real machine.

**Warning: the job can take several hours before launching, please estimate your time accordingly!**

(e) Shortly comment on the nature of fluctuations you observe for the ideal simulator, the noisy simulator, and the real machine.