DMC is Discrete Mathematics and Computing (orange textbook). Problems are also on the class website.

Always show your work and explain your reasoning.

Problem numbers refer to the version of the chapter problems on the course webpage.

The point value in parentheses are for the 6xxx version of the class.

ASSIGNMENT 6

[100 (100)] Open ended. Work in teams of 1, 2, or 3

Build a quantum computer. Well, okay no. Build a classical simulator for a quantum computer.
Your task is to build a "programming interface" for quantum programming and a classical simulation of the quantum circuit that the user "programs". Here are the capabilities your program should have:

(1) Programmer can specify n the number of input qubits, for example $n = 4$.

(2) Programmer can specify any number of levels of quantum computation. Let ℓ be the number of levels.

(3) At each level of the quantum computation, the programmer can specify what happens to each qubit and which qubits are controlling. The qubits could be acted on by standard quantum gates or by a classical Boolean function f represented by its unitary operator U_f.

You may create your own "quantum programming language". Here is an illustrative example program.

$n = 4$
Level 1: $\{q_2, \text{NOT}, q_1\} \mid \{q_1, H, q_2\ q_3\}$
Level 2: $\{q_1, H, \text{NULL}\} \mid \{q_2, q_3, q_4, f, q_1\}$

The number of levels is implicit, $\ell = 2$. Let's parse this program. Each 3-tuple has the format:

$$\{\text{bit or bits being operated on, operator, controlling bits}\}$$

In level 1, NOT is applied to q_2 controlled by q_1, which means not is applied to q_2 if $q_1 = 1$ otherwise nothing is done to q_1. And, H is applied to q_1, controlled by q_2, q_3. This means if H is applied to q_1 if $q_2 = q_3 = |1\rangle$ and otherwise nothing is done to q_1.

In level 2, H is applied to q_1 and the function f is applied to (q_2, q_3, q_4), controlled by q_1. This means f is a function on 2 bits and U_f is applied on $|q_2\rangle \otimes |q_3\rangle \otimes |q_4\rangle$. The function f must also be defined somewhere, for example before the program starts, or in a separate file of function definitions which can be loaded into (say) python as a library. Note, the program can also be read by python and parsed.

Your goal is to write a program (say in python) that interprets the quantum-program and then computes the operator for the circuit and applies the operator on a pure state, for example the state $|0, 1, 1, 0\rangle$ to produce an output state ψ. The user can then input a set of bit indices for example $\{2, 4\}$ and the program should report $\mathbb{P}[q_2 = 1, q_4 = 1]$.

Grading. Contest in the last class. I will give a 10-qubit circuit, an input pure state and which qubits are measured. Each team-member programs the circuit independently in class, runs on the input and produces the probability. Files handed into submitty: your quantum computer simulator, your program and function-def file, and a file containing the probabilities for the possible outcomes of the measured qubits.
Your score will be based on the time it took for you to upload the correct probability.