Project B1: **Custom Quantum Architecture**

Develop a customized quantum computer to perform a specific computation.

Examples: Quantum Fourier Transform, modular exponentiation, ...
Something non-trivial that is a building block for quantum applications

**Constraints**
- Set "realistic" constraints for near-term quantum hardware
- Can speculate on the invention of a new device / interconnect, within reason
- E.g., don't assume full connectivity among qubits

**Metrics and Evaluation**
- How will you measure the cost & performance of your design?
- Assuming qiskit -- may need to modify simulator, software stack

2-3 people
multiple teams allowed if choose computations
Project B2: **Combinatorial Optimization**

Choose a combinatorial optimization problem and solve it using the gate model (QAOA or VQE) and annealing (D-Wave)

Examples: SAT, graph coloring, traveling salesman, knapsack, ...

The hard part...

Choosing a reasonable problem given limited qubits

Determining the Hamiltonian for your problem

Evaluation

Compare and contrast solutions on IBM Q and D-Wave Programming, tools, quality of answers, etc.

2 people

multiple teams allowed if different problems
Project B3: Ancilla Qubit support in Qiskit

Add support in qiskit for managing a "pool" of ancilla qubits, to be recycled across various parts of the larger circuit.

Motivation
- Multiple functions need ancilla qubits
- Each creates a set and adds to the quantum circuit -- cannot be removed
- Adds more and more qubits to the circuit instead of reusing
  (Or ancilla qubits have to be exposed to the upper-level application)

Implementation and Demonstration
- Requires understanding and modification of qiskit infrastructure
- Show ancillas being created "on demand" and recycled if available
- Support for uncomputation within a function

1 person
Project B4

Find one optimization/numerics/machine learning algorithm from those listed on https://quantumalgorithmzoo.org/#ONML. (1) Implement it on IBM Qiskit and run it with limited problem sizes on both simulators and IBM quantum computers. (2) Optimize your implementation to improve the success rate when running on the IBM quantum computers.

Team size: 1-2 depending on the complexity of the algorithm selected.