

QAMP 2021 – Checkpoint 1

Saasha Joshi

Project: Good first issues in networkx

Mentor: Matthew Treinish



Deliverables

Understanding Rust and Graph Theory

- Getting started with Rust.

Networkx code base is written primarily (~80-90%) in Rust.

- Python API is built using PyO3.
- Understanding basic graphs and graph theory to build networkx modules.

GitHub source code and Networkx

- Networkx built to replace Qiskit's dependence on Networkx – a Python graph library.
- Building from source and relying on the network documentation for reference.

Good First Issues

- Solve issues such as
 - expanding the generators module,
 - expanding graph operations,
 - adding new layout methods etc.

Progress – Checkpoint 1

- Learning Rust.
- Understanding basic graphs and graph theory.
- Referring the network documentation for code initiation.
 - Lattices
 - Geometric graphs
 - Graph Atlas
 - Intersections

```
from qiskit import QuantumCircuit, execute
from qiskit import Aer, IBMQ
from qiskit.providers.aer.noise import NoiseModel

# Choose a real device to simulate from IBMQ provider
provider = IBMQ.load_account()
backend = provider.get_backend('ibmq_vigo')
coupling_map = backend.configuration().coupling_map

# Generate an Aer noise model for device
noise_model = NoiseModel.from_backend(backend)
basis_gates = noise_model.basis_gates

# Generate 3-qubit GHZ state
num_qubits = 3
circ = QuantumCircuit(3, 3)
circ.h(0)
circ.cx(0, 1)
circ.cx(1, 2)
circ.measure([0, 1, 2], [0, 1, 2])

# Perform noisy simulation
backend = Aer.get_backend('qasm_simulator')
job = execute(circ, backend,
              coupling_map=coupling_map,
              noise_model=noise_model,
              basis_gates=basis_gates)
result = job.result()
print(result.get_counts(0))
```

