Homework 4

Due On: April 22, 2021 3:30PM (CST)

Quantum Walk on a Line: Recall the notion of a classical (random) walk on an infinite line graph. Vertices are labeled according to $\mathbb{Z}$: $\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$ where vertices labeled i and $i+1$ are adjacent. We start our walk at the origin and in each step with probability $\frac{1}{2}$ either move to the left or the right neighbor. Let $N > 0$ be the length of the walk W_N. Then W_N induces a probability distribution $p_N : \mathbb{Z} \to [0,1]$ where $p_N(i)$ is the probability that our walk ends at vertex i after N steps. It is well known that as $n \to \infty$, $p_N \to \mathcal{N}(0, N)$, where $\mathcal{N}(0, N)$ is the normal distribution with zero mean and standard deviation of $\sqrt{N}$. This tells us that the probability of us visiting a vertex which is at distance $d = \Omega(N)$ from the origin is exponentially small (in d). Hence we need a walk of length about $O(N^2)$ to guarantee with constant probability that we visit every vertex within $O(N)$ distance from the origin.

Can we do better using Quantum Mechanics? It turns out the answer is yes. Just like the Grover's search algorithm, we gain a quadratic speed-up in the quantum version (this is not a coincident). In the classical walk described above we toss a fair coin at each step and based on the outcome we proceed to either the left or the right neighbor. Two explain quantum walks we need to introduce a notion of a quantum coin, which is realized by a qbit. Classically, the state of the walk is just the current position. In the quantum case, we describe the state as a tensor product $|v\rangle|c\rangle$ where $|v\rangle$ is the vertex state (encoding our current position as a superposition) over the basis $\{|n\rangle \mid n \in \mathbb{Z}\}$ and $|c\rangle$ corresponds to the coin state qbit. The transition operations are then defined in terms of unitaries as follows. Each step of the walk involves two operation, one acting on the coin state (U_c) and the other on the vertex state (U_v). Here U_c can be any single qbit operations. In this homework we will consider two specific examples, H and $\frac{1}{\sqrt{2}}(I + iX)$. Now we define U_v.

$$U_v |n\rangle|0\rangle = |n-1\rangle|0\rangle \text{ and}$$
$$U_v |n\rangle|1\rangle = |n+1\rangle|1\rangle$$

Each step consists of applying $U = U_v(I \otimes U_c)$ to the current state $|v\rangle|c\rangle$. We start at the origin and hence we can write our initial state as $|0\rangle|c_0\rangle$. Here we assume the quantum coin may be in some arbitrary initial state (This is a departure from the classical case where we assume the outcome did not depend on the state of the coin). A quantum walk Q_N of length N is equivalent to applying U^N to $|0\rangle|c_0\rangle$. At this point if we measure the first register we get the final position. As it turns out based one the choice of $|c_0\rangle$ and U_c the final probability distribution can be remarkably different.

Analysis: We can measure the final vertex state $|v\rangle$ using the ON basis $\{|n\rangle \mid n \in \mathbb{Z}\}$ and determine the probability $q_N(n)$ of ending in position n after N steps. Notice the this basis is infinite. In a computer everything is finite. So we consider a subset by noting that we can only travel up a vertex at a distance N from the origin in N-steps. Hence we may restrict our space to be the ON basis $\{|n\rangle \mid n \in \{-N, \ldots, N\}\}$. This will allow us to specify any arbitrary vertex state as follows:

$$|v\rangle = \sum \alpha_n |n\rangle$$

where, $\sum_{n\in\{-N,\ldots,N\}} |\alpha_n|^2 = 1$. To derive a recursive formula for the next state based on the current state, let us write the state after i steps as $|\psi_i\rangle = \sum \alpha_n^i |n\rangle |0\rangle + \sum \beta_n^i |n\rangle |1\rangle$. Let $U_c = \begin{bmatrix} \gamma_0 & \gamma_1 \\ \delta_0 & \delta_1 \end{bmatrix}$. Then after applying U_c we have:

$$|\psi_i'\rangle = U_c |\psi_i\rangle = \sum(\alpha_n^i \gamma_0 + \beta_n^i \gamma_1) |n\rangle |0\rangle + \sum(\alpha_n^i \delta_0 + \beta_n^i \delta_1) |n\rangle |1\rangle$$

Now we apply U_v and get:

$$\begin{aligned} |\psi_{i+1}\rangle = U_v |\psi_i'\rangle &= \sum(\alpha_n^i \gamma_0 + \beta_n^i \gamma_1) |n-1\rangle |0\rangle + \sum(\alpha_n^i \delta_0 + \beta_n^i \delta_1) |n+1\rangle |1\rangle \\ &= \sum(\alpha_{n+1}^i \gamma_0 + \beta_{n+1}^i \gamma_1) |n\rangle |0\rangle + \sum(\alpha_{n-1}^i \delta_0 + \beta_{n-1}^i \delta_1) |n\rangle |1\rangle \end{aligned}$$

Hence, $\alpha_n^{i+1} = \alpha_{n+1}^i \gamma_0 + \beta_{n+1}^i \gamma_1$ and $\beta_n^{i+1} = \alpha_{n-1}^i \delta_0 + \beta_{n-1}^i \delta_1$.

The above recursive formulation gives you a way to compute the amplitudes of the successive states in the walk. Even though, it does not effect the result we identify $|N+1\rangle$ with $|-N\rangle$ and $|-N-1\rangle$ with $|N\rangle$ in the above calculation [1].

Implementation We will pick our initial state $|c_0\rangle$ from the set $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ and U_c from $\{H, \frac{1}{\sqrt{2}}(I + iX)\}$. Hence there are 8 possible combinations for $(|c_0\rangle, U_c)$. Write a Python program to simulate the quantum walks according to the specification described above. You should present the final probability distribution as a histogram for each of the 8 possible setups. Take $N = 1000$ and run the walk for N steps.

[1] We have this edge case due to our walk being finite.