Levon Keshishoghlyan | Final Project
CS 339 Introduction to Quantum Computing | 2024-05-17

Description of the Problem

Consider the famous puzzle game Master Mind played by two players. Given n different colors, the first player - the keeper, secretly forms a sequence of n colored pins, where several pins may share the same color, or all of them may be of different colors. The task of the second player – the guesser, to disclose the hidden sequence with minimal guesses. Work on any of the following two versions:

- **Version 1:** each guess is graded by the keeper with two digits – the first being the number of correct pins in their correct positions and the second being the number of pins with correct colors but in wrong places. The game stops when the grade of the most recent guess is n_0.
- **Version 2:** each guess is graded by the keeper with a single digit – the number of correct pins in their correct positions. The game stops when the grade of the most recent guess is n.

Solution used

I've taken version 2 which returns only one number of correctly predicted positions. I have also looked at each position separately as the correctness of a single position does not depend on the others and it's obvious that for a secret sequence $s \in n^n$, prediction $x \in n^n$ and function $F(x, s) = |i \in 1, 2, \ldots, n : s_i = x_i|$ we can separate function F into:

$$F(x, s) = F_0(x_0, s_0) + F_1(x_1, s_1) + \cdots + F_n(x_n, s_n)$$

where $F_i(x_i, s_i) = 1$ when $x_i = s_i$ and 0 otherwise.
All of the following solutions will rely on the parallelization of the sequence finding between n separate sequences running for each position.

Task 1

The link of the repo was shared by the preliminary date

Task 2

The time complexity of the algorithm in the classical case will be $O(n)$ for each process as the algorithm will need to check each possible color option one by one. The quantum algorithm will perform the same task in $O(\sqrt{n})$ which will be discussed in the last task. As each process for a position runs the same task the overall time complexity for the whole sequence won't differ.

Task 3

To describe color in a position we will need $\lceil \log_2 n \rceil = k$ bits/qubits. For the classical algorithm, one additional 0 bit will be needed to check the result and for the quantum algorithm, one additional qubit will be needed in the $|-\rangle$ state for the phase kick-back. The qubits in the quantum algorithm will also need to be in a superposition, therefore $H^{\otimes k}$ will be applied to them.
To summarize for the classical algorithm we will simply encode each color to the respective binary version of it and pass it to the algorithm. The initial vector will be x_i. The quantum algorithm on the other hand will initialize k zeros and then put them into superposition using Hadamard gates and will get $|\phi\rangle = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle$

Task 4

To implement a classical algorithm we will simply need a transformation $U_f |x_i\rangle |0\rangle = |x_i\rangle |0 \oplus f(x_i)\rangle$ where $f(x_i) = 1$ when $x_i = s_i$ and 0 otherwise. Then using controlled-U_f gate (it's important to note that for the classical case, we will need some NOT gates for the correct key matching) we can change the value of the second register to 1 when we check for the correct $|x_i\rangle$. We just measure the second register after each initialization and if it's 1 we stop the algorithm for that process, on the other hand, if it's 0 we apply the same algorithm but with $|x_i \oplus 1 \mod n\rangle$. In case if second register stays 0 for the n times then we can conclude that the keeper has cheated somewhere.

Task 5

To implement the quantum algorithm we will use Grover's search algorithm. The controlled qubit will be $|-\rangle$ as mentioned above and we will need one more subroutine, namely the diffusion operator. We will simply need to implement $H^{\otimes k}$ one more time then apply controlled-$R_0 = 2\,|0^{\otimes k}\rangle\,\langle 0^{\otimes k}| - \mathrm{I}$ and $H^{\otimes k}$ one more time to come to the regular basis. The whole process of applying controlled-U_f then diffusion will then be repeated $\lceil \frac{\pi}{4} n \rceil$ times to bring the probability of getting a correct result closer to 1.

Example

Look for an example with $k = 5$ in the attached jupyter notebook.