In the beginning all states in a qubit system are equally probable, then we make the first guess, and reduce the number of possible candidates for an answer with the help of Grover's algorithm, that is decreasing the probabilities of some states. The process continues until we are left with a single state which will be the answer.

The best case scenario is when we guess the correct number on the first try, or if we guess its complement (in this case we win the game in 2 steps).

No matter what the first guess is *(see the tables in CS231_Final_Project_AY)* there are some values that can be excluded from the list of candidates for our second guess based on some logic/connection between them. For example, if our hidden number is 1011 we can identify the repeating pattern of 1 2 2 3 with a certain periodicity, 3 to be precise.

This was the case, when considering a certain hidden number. In the second table on the sheet I have highlighted some patterns for each of the candidates. Note that some values repeat if we consider them based on modulo 2. Probably, the pattern finding process can be completed with the help of Fourie Transform.

Apparently, the data that has to be stored is the following: the previous guess, potential next guess and the number of matching digits of the guess.

The potential next guess is determined by estimating the superposition created by the first guess. Based on the number of matching digits , i.e. grade, we separate "good" and "bad",sets of potential guesses, then adjust the probabilities of "good" guesses to be equal.

In case of using quantum computing for the first n guesses, we create n superpositions with equal probability for each outcome. After grading the first guess, we separate it into good and bad states and do a measurement to get the second guess. Using the information on the first superposition's good and bad states, we adjust the second superposition state to be the same as the first one before measurment. By applying another gate to the second superposition based on the grading of the second guess, we generate new "good" and "bad" states from the previous "good" states. Repeating this process n times significantly reduces the number of possible candidates.

We can introduce a function for keeping track of the previous move's potential superposition for the next move, as well as the grade of the previous move. This function needs to output the next move by separating good and bad states and performing the measurement. The function will be used continuously until the correct guess is reached.