Qubits at work – from codebreaking to climate modelling

Worksheet – solutions

Exercise 1: Understanding Grover's algorithm

Objective: To get a basic understanding of how Grover's algorithm works and how it can speed up a search in an unsorted list of items. You need to be a group of five students to carry out this exercise.

Step 1: Setting up the search problem

- Create a list of four items (these can be numbers, letters or objects, e. g. A, B, C and D).
- Choose one of the items as the hidden item to be found, for instance item C.

Step 2: Simulating a classical search

1. **Role-play**
   One student plays the role of a user of a classical computer looking for a hidden item. The other four students are the items A, B, C, D. The four students know who represents the hidden item, the "user" does not.
2. **Query process**
   The "user" points to one classmate at a time and asks: "Are you the correct item?" The students reply with **"Yes"** or **"No".**
3. **Record the number of queries**
   Keep track of how many times the question has to be asked to get the answer "Yes".
4. **Discuss the three cases for *N* = 4.**

Case	What happens	Number of queries needed
Best	The first guess is correct.	**1**
Worst	The item is the last one checked.	**4**
Average	Each of the four positions is equally likely, so the average number of queries is approximately $N / 2 = 4 / 2 = 2$. In average, the user will have to ask the question **twice**.	**2 in average**

Step 5: Discussion & questions

- **How does the oracle differ from a classical "hint"?**

  It checks all the items at once (without looking) and marks the hidden item, but doesn't get an answer like "this is not the hidden item".

- **What does the amplification step do?**

  It decreases the probability of the unmarked items and increases the probability of the marked item to be the correct answer. The hidden item stands out as the probable answer more quickly than when using sequential queries in a classical search.

- **Describe the best, worst and average case when using a classical search to find an item among 12 unsorted items.**

Case	What happens	Number of queries
Best case	You pick up the very first item and it is the one you want.	**1**
Worst case	You have to look at every single item because the target is the last one you examine.	**12**
Average case	The target is equally likely to be in any position, so on average you look halfway through the list. Mathematically, that is 12/2 = 6.	**6 on average**

So, a classical search can be very quick if you have to ask only once because the hidden item is the first one in the list. But you might also have to ask 12 times (in fact, 11 times is enough, because once you have the answer to the 11^{th} query, you know whether the item is in position 11 or 12), which will take a while. In average, 6 queries are needed to find the hidden item.

- **Verify whether the average number of queries is 6 (= *N*/2) with the method described in the box above.**

  The possible cases are:
  Case 1 – yes no no no no no no no no no no no → requires 1 query = the first student is the hidden item (**yes**), the other 11 students are not (**no**).
  Case 2 – no yes no no no no no no no no no no → requires 2 queries.
  Case 3 – no no yes no no no no no no no no no → requires 3 queries.
  Case 4 – no no no yes no no no no no no no no → requires 4 queries.
  Case 5 – no no no no yes no no no no no no no → requires 5 queries.
  Case 6 – no no no no no yes no no no no no no → requires 6 queries.
  Case 7 – no no no no no no yes no no no no no → requires 7 queries.
  Case 8 – no no no no no no no yes no no no no → requires 8 queries.
  Case 9 – no no no no no no no no yes no no no → requires 9 queries.
  Case 10 – no no no no no no no no no yes no no → requires 10 queries.
  Case 11 – no no no no no no no no no no yes no → requires 11 queries.
  Case 12 – no no no no no no no no no no no yes → no additional query is needed, because if you know that the 11^{th} item is **no**, then you know that the 12^{th} item is **yes.**

To calculate the average number of queries, one has to add up all the numbers of required queries and divide by 11 (i. e. the number of possible cases; case 12 can be discarded, it is redundant to case 11):
(1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11) / 11 = 66 / 11 = 6;
As a result, the average number of required queries is: ***N*/2 = 12 / 2 = 6.**

- **How many steps would be required to find the item using Grover's algorithm?**

  It is very likely that you will find the correct item after $\sqrt{12} \approx 3.5$ – so, between 3 and 4 "oracle + amplification" steps.