Thurs: HW - project report

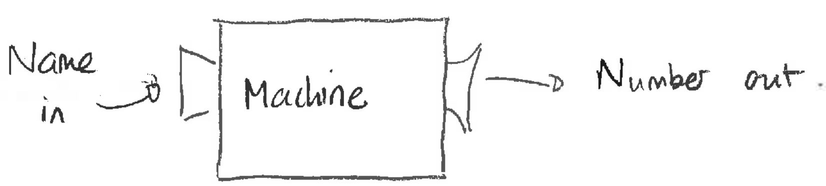
Database search

A database search problem involves a database for which one or more items are "marked". The task is to locate a "marked" item

For example consider a rudimentary phone book

Name	Number
Alice	4
Bob	6
Charlie	1
Dan	8
Eve	2

We can easily provide a number given a name. So we have a machine:



The machine can easily operate in a forwards direction. But reverse is supposedly harder.

Thus, if given a number and we are asked to find the associated name then the machine does not work in reverse. So suppose we are given the question:

"Find the person whose number is 8."

We would search from top to bottom and provide various inputs to the machine

Alice $\to$ Mach. $\longrightarrow$ 4 $\Rightarrow$ Not Alice

Bob $\to$ Mach. $\longrightarrow$ 6 $\Rightarrow$ Not Bob

$\vdots$

Dan $\to$ $\square$ $\longrightarrow$ 8 $\Rightarrow$ Yes Dan.

Here Dan is the "marked item" that we have uncovered.

We need to convert this into a more mathematical operation and can illustrate this in various ways.

<u>Example:</u> (Factorizing as searching)

Given an integer Z we aim to find a factor of Z. Possible candidates are all primes $2, 3, 5, 7, \ldots$. These make the database "entries". The factors of x are the "marked" items

Database	
2	
3	
5	
7	
11	
13	
$\vdots$	

We can locate a marked item by, one at a time, dividing z by each database entry. We can perform

divide z by any database entry x

If x divides z then output "1"

If x does not divide z then output "0".

So we could define an oracle function

$$f(x) = \begin{cases} 0 & \text{if } x \text{ does not divide } z \\ 1 & \text{if } x \text{ '' divide } z \end{cases}$$

For this particular procedure there are efficient classical methods for constructing an oracle that can check if x divides z. So

With this oracle it is easy to check if a given database entry is or is not marked but it is difficult to find such a database entry

So if $z = 247$ the database + oracle would give

x	f(x)
2	0
3	0
5	0
7	0
11	0
13	1
17	0
19	1
23	0
⋮	⋮
	0

We would search the database by evaluating f successively

$$f(2) = ?$$

$$f(3) = ?$$

$$f(5) = ?$$

until we obtain $\quad f(?) = 1$.

This gives a general formulation of the database search problem.

1) A database consists of N possible locations

$$x = 0, 1, 2, \ldots, N-1$$

and some of which are "marked"

2) A database is equipped with an oracle function

$$f(x) = \begin{cases} 0 & \text{if } x \text{ is not "marked"} \\ 1 & \text{if } x \text{ is "marked"} \end{cases}$$

The database function is easy to evaluate classically

3) A marked item can only be located by oracle queries.

The task is:

$$\boxed{\text{Locate at least one marked item with minimal oracle queries}}$$

1 Database oracles

Consider a database whose oracle functions are given, in terms of functions mapping multiple bits to a single bit, as below. For each determine the size of the database, list the possible database entries and find a marked item.

a) $f_1(x_1, x_0) = x_1 x_0$

b) $f_2(x_1, x_0) = x_1 x_0 \oplus x_0$

c) $f_3(x_1, x_0) = x_1 x_0 \oplus x_1$

d) $f_4(x_1, x_0) = x_1 x_0 \oplus x_1 \oplus x_0 \oplus 1$

e) $f_5(x_2, x_1, x_0) = x_2 x_1 x_0$

f) $f_6(x_2, x_1, x_0) = x_2 x_1 x_0 \oplus \cancel{x2}x0$ $x_2 x_0$

Now consider f_1, f_2, f_3, f_4.

g) If you were given one of these, how many oracle queries are needed to locate the marked item with certainty?

h) If you were given one of these, and each is equally, likely, how many oracle queries do you need on average to locate the marked item?

Answer: a) b) c) d) each is a function of two bits

$\Rightarrow$ 4 possible database entries.

x_1	x_0	f_1	f_2	f_3	f_4
0	0	0	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	1	0	0	0

for f_1 marked is 11

f_2 " is 01

f_3 " is 10

f_4 " is 00

e) f) 3 bits $\Rightarrow$ 8 database entries.

x_2	x_1	x_0	f_5	f_6
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0

For f_5 marked at 111

f_6 " " 110

g) For certainty we would need to evaluate

$f(00)$
$f(01)$
$f(10)$

since ~~if~~ there is at least one marked item ...

h) average # = $\frac{1}{4}$ number for f_1 + $\frac{1}{4}$ number for f_2 + ...

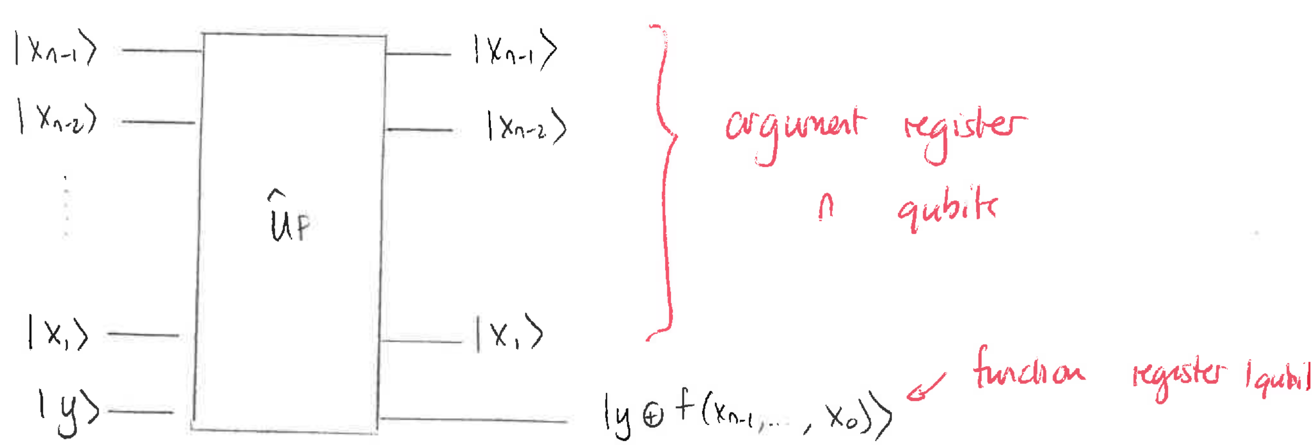
$$= \frac{1}{4}(3+2+3+1) = \frac{9}{4}$$

This illustrates a general result

> If a database is unsorted or unstructed and contains one marked item out of N then
>
> - to find the marked item with certainty requires $N-1$ oracle queries
>
> - on average the number of oracle queries is $\frac{N-1}{2}$

Quantum oracles for database search

As before we will construct a unitary operation that allows for evaluating the oracle function. Specifically for a database requiring n bits (i.e. $N = 2^n$) the oracle maps



or

$$\hat{U}_F |x_{n-1} \dots x_0\rangle |y\rangle \to |x_{n-1}, \dots x_0\rangle |y \oplus f(x_{n-1}, \dots, x_0)\rangle$$

At this point it is preferable to use decimal notation for the qubit labels. So for the binary number

$$x_{n-1}, x_{n-2} \cdots x_1 x_0$$

if the decimal representation is x. This requires n qubits and we write

$$|x\rangle_d \equiv |x_{n-1} \ldots x_0\rangle$$

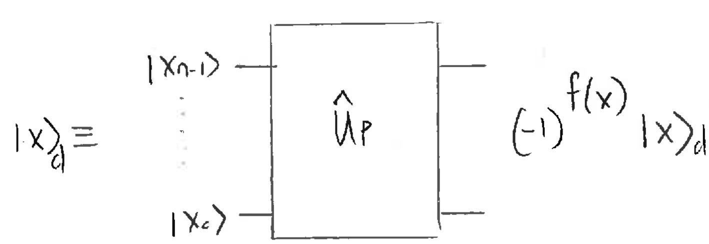
where "d" means that x is a decimal number. This is still an n qubit state. So

$$\hat{U}_f\, |x\rangle_d |y\rangle = \underbrace{|x\rangle_d}_{n \text{ qubits}} \underbrace{|y \oplus f(x)\rangle}_{1 \text{ qubit}}$$

Now as before, we can show that if the function register is initially in the state $(|0\rangle - |1\rangle)/\sqrt{2}$ then

$$\hat{U}_f\, |x\rangle_d \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = (-1)^{f(x)} |x\rangle_d \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

With this strategy the function register becomes irrelevant. Thus we can consider the modified oracle



or

$$\hat{U}_f\, |x\rangle_d = (-1)^{f(x)}\, |x\rangle_d$$

2 Search oracle

Consider the oracle function

$$f(x_2, x_1, x_0) = x_2 x_1 x_0 \oplus x_2 x_0 \oplus x_1 x_0 \oplus x_0.$$

and the corresponding oracle unitary operator defined via

$$\hat{U}_f \left| x \right\rangle_d = (-1)^{f(x)} \left| x \right\rangle_d.$$

For the given function determine the effect of the oracle on all basis states $\left|0\right\rangle_d, \left|1\right\rangle_d, \left|2\right\rangle_d, \left|3\right\rangle_d, \ldots$.

Answer:

decimal x	binary x_2	x_1	x_0	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

$|0\rangle_d \to |0\rangle_d$ $\qquad$ $|4\rangle_d \to |4\rangle_d$

$|1\rangle_d \to |1\rangle_d$ $\qquad$ $|5\rangle_d \to |5\rangle_d$

$|2\rangle_d \to |2\rangle_d$ $\qquad$ $|6\rangle_d \to |6\rangle_d$

$|3\rangle_d \to -|3\rangle_d$ $\qquad$ $|7\rangle_d \to |7\rangle_d$.

Note that the component of basis vector corresponding to the marked location is tagged with (-1).