A quantum algorithm

Ryan LaRose
ISE 870
April 22, 2019

Orientation & Learning goals

Where are we?

➔ Lecture 11

Lecture 11	Deutsch-Jozsa algorithm	DJ problem, classical and quantum complexity, quantum algorithm, experiment

What you should (hopefully) know by this point.

➔ Qubits.
➔ Basic quantum gates.

Learning goals: Be able to...

- Define the DJ problem.
- Explain the query complexity of DJ in the classical setting.
- Construct a quantum algorithm with smaller query complexity than the classical case.

Defining the DJ Problem: Background

Let f be a function from one bit (0 or 1) to one bit (0 or 1).

That is, f inputs one bit (0 or 1) and outputs one bit (0 or 1).

Q: How many different functions f can there be?

Defining the DJ Problem: Background

Let f be a function from one bit (0 or 1) to one bit (0 or 1).

That is, f inputs one bit (0 or 1) and outputs one bit (0 or 1).

Q: How many different functions f can there be?

A: Four!

Here's a table listing all possible functions:

x	f_0	f_1	f_x	$f_{\bar{x}}$
0	0	1	0	1
1	0	1	1	0

Exercise: Draw graphs for each function.

Defining the DJ Problem: Background

We can divide these functions into two “classes” or groups.

Defining the DJ Problem: Background

We can divide these functions into two “classes” or groups.

Constant functions have the same output *regardless of the input*.

i.e., the output is independent of the input, or *constant* in the input.

Defining the DJ Problem: Background

We can divide these functions into two “classes” or groups.

Constant functions have the same output *regardless of the input*.

i.e., the output is independent of the input, or *constant* in the input.

Balanced functions have different outputs for each input.

i.e., the output depends on the input.

Defining the DJ Problem: Background

We can divide these functions into two “classes” or groups.

Constant functions have the same output *regardless of the input*.

i.e., the output is independent of the input, or *constant* in the input.

Balanced functions have different outputs for each input.

i.e., the output depends on the input.

Question: Which functions that you drew are constant? Which are balanced?

Defining the DJ Problem

I have a function *f* from one bit to one bit. You don't know which one it is.

Defining the DJ Problem

I have a function f from one bit to one bit. You don't know which one it is.

You want to know: Is it **constant** or **balanced**?

Defining the DJ Problem

I have a function f from one bit to one bit. You don't know which one it is.

You want to know: Is it **constant** or **balanced**?

However...

Defining the DJ Problem

I have a function *f* from one bit to one bit. You don’t know which one it is.

You want to know: Is it **constant** or **balanced**?

However...

I’m a mean guy.

Defining the DJ Problem

I have a function *f* from one bit to one bit. You don't know which one it is.

You want to know: Is it **constant** or **balanced**?

However...

I'm a mean guy.

I won't tell you directly if it's constant or balanced.

Rules: You can only ask me what the value of f is at one particular input at a time.

Example: Playing the DJ game

I’ll be the “oracle.”

I’m thinking of some function, you want to know whether it’s constant or balanced.

Ask me questions!

Activity: Playing the DJ game

Now play the DJ game at your table.

Select one person to be the “oracle.”

Everyone else asks questions.

Record how many questions it takes to determine whether the function is constant or balanced.

Switch so that everyone gets a chance to be the oracle.

How many questions to win the DJ game?

How many questions to win the DJ game?

Exactly **two**!

Here’s a “formal proof:”

How many questions to win the DJ game?

Exactly **two**!

Here's a "formal proof:"

We want to determine the *parity*

$$f(0) \oplus f(1)$$

How many questions to win the DJ game?

Exactly **two**!

Here's a "formal proof:"

We want to determine the *parity*

$$f(0) \oplus f(1)$$

Either f(0) or f(1) could change the value of the parity, and they are independent of each other.

How many questions to win the DJ game?

Exactly **two**!

Here's a "formal proof:"

We want to determine the *parity*

$$f(0) \oplus f(1)$$

Either f(0) or f(1) could change the value of the parity, and they are independent of each other.

Since we can only learn ONE of f(0) or f(1) per question, we must ask two questions.

How many questions to win the DJ game?

Exactly **two**!

Here's a "formal proof:"

We want to determine the *parity*

$$f(0) \oplus f(1)$$

Either f(0) or f(1) could change the value of the parity, and they are independent of each other.

Since we can only learn ONE of f(0) or f(1) per question, we must ask two questions.

Question: How can determining the parity tell us if the function is constant or balanced?

The quantum case

A computer scientist would say that “the classical *query* complexity of DJ is two.”

Let’s now give ourselves access to a quantum computer.

The quantum case

A computer scientist would say that “the classical *query* complexity of DJ is two.”

Let’s now give ourselves access to a quantum computer.

Question: What do you think the *quantum query complexity* of the DJ problem is?

Answer: Just one!

The quantum case

A computer scientist would say that “the classical *query* complexity of DJ is two.”

Let’s now give ourselves access to a quantum computer.

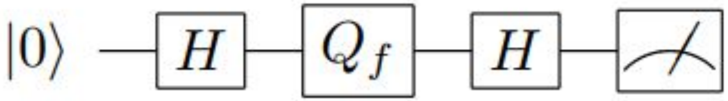
Question: What do you think the *quantum query complexity* of the DJ problem is?

Answer: Just one!

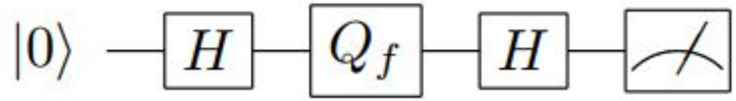
That is, we can win the DJ game by asking only one question.

Which question should we ask?

The solution?!

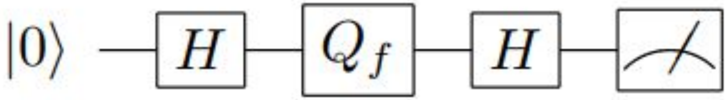


The solution?!



Uhhh… what does that mean?

Key points

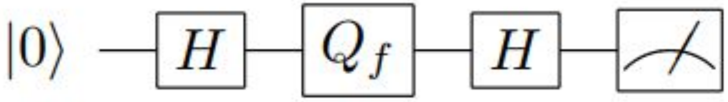


We're asking **only one** question. (There's one Q_f.)

We use **superposition** to ask a different type of question which is not possible in the classical world.

Let's now break this down to understand it.

Key points



We’re asking **only one** question. (There’s one Q_f.)

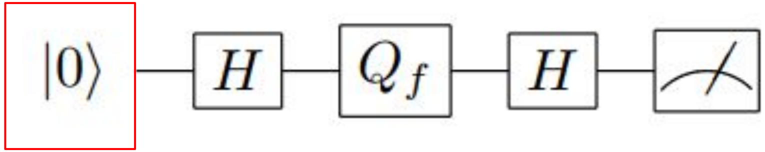
We use **superposition** to ask a different type of question which is not possible in the classical world.

Let’s now break this down to understand it.

The class would have had ten lectures on background for this diagram!

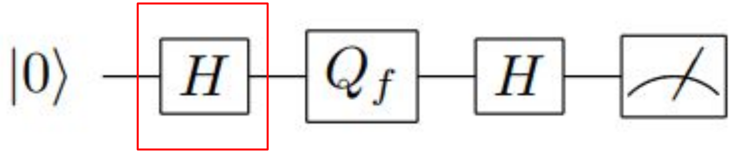
It’s a lot to take in at once, but should be understandable.

Understanding the DJ algorithm



Far left: The **initial state** is the |0> qubit.

Understanding the DJ algorithm

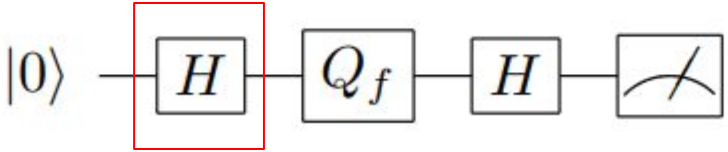


Far left: The **initial state** is the |0> qubit.

Step two: Hadamard gate

$$H|0\rangle = |0\rangle + |1\rangle$$
$$H|1\rangle = |0\rangle - |1\rangle$$

Understanding the DJ algorithm



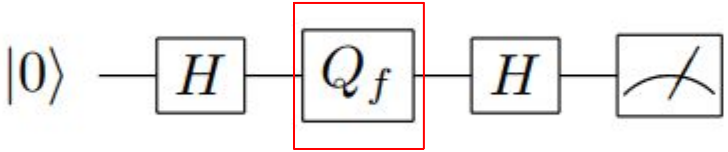
Far left: The **initial state** is the |0> qubit.

Step two: Hadamard gate

$$H|0\rangle = |0\rangle + |1\rangle$$
$$H|1\rangle = |0\rangle - |1\rangle$$

Question: What is the state of our qubit after the first Hadamard gate?

Understanding the DJ algorithm

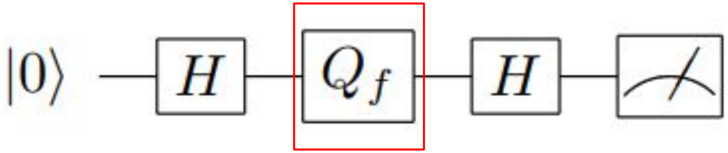


Step three: Query the oracle. (Ask the question.)

[Through things we won't have time to discuss] This results in the state:

$$(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle$$

Understanding the DJ algorithm



Step three: Query the oracle. (Ask the question.)

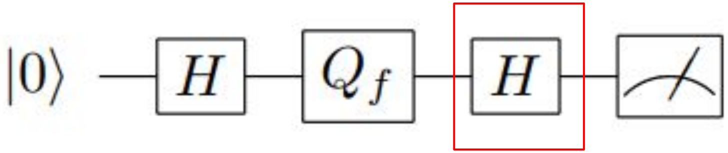
[Through things we won't have time to discuss] This results in the state:

$$(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle$$

Key: We're still asking **only one** question!

We use **superposition**, a feature of quantum computers, to send in a different input.

Understanding the DJ algorithm



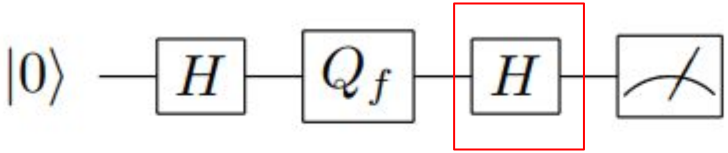
Step four: Another Hadamard!

Challenge Q: Using the fact that H is *linear,* write down the state of our qubit after the second Hadamard gate.

Note: *Linear* means that

$$H(\alpha|0\rangle + \beta|1\rangle) = \alpha H|0\rangle + \beta H|1\rangle$$

Understanding the DJ algorithm



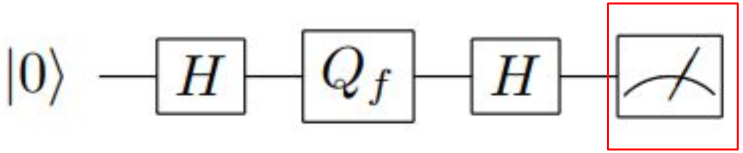
Step four: Another Hadamard!

Challenge Q: Using the fact that H is *linear,* write down the state of our qubit after the second Hadamard gate.

Answer:

$$\left[(-1)^{f(0)} + (-1)^{f(1)}\right] |0\rangle + \left[(-1)^{f(0)} - (-1)^{f(1)}\right] |1\rangle$$

Understanding the DJ algorithm



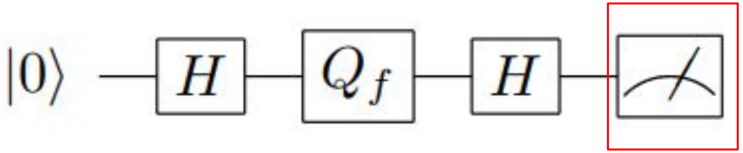
Step five: Understanding the measurement.

Quantum fact: The detector returns 0 if the state is |0> and 1 if the state is |1>.

Question: Suppose f is constant (which means f(0) = f(1)). What will our detector measure?

$$\left[(-1)^{f(0)} + (-1)^{f(1)}\right] |0\rangle + \left[(-1)^{f(0)} - (-1)^{f(1)}\right] |1\rangle$$

Understanding the DJ algorithm



Step five: Understanding the measurement.

Quantum fact: The detector returns 0 if the state is |0> and 1 if the state is |1>.

Question: Suppose f is balanced (which means f(0) != f(1)). What will our detector measure?

$$\left[(-1)^{f(0)} + (-1)^{f(1)}\right]|0\rangle + \left[(-1)^{f(0)} - (-1)^{f(1)}\right]|1\rangle$$

Conclusions and takeaways

We measure 0 if and only if f is constant.

We measure 1 if and only if f is balanced.

Conclusions and takeaways

We measure 0 if and only if f is constant.

We measure 1 if and only if f is balanced.

Running the DJ quantum circuit tells us whether f is constant or balanced with only one question.

Conclusions and takeaways

We measure 0 if and only if f is constant.

We measure 1 if and only if f is balanced.

Running the DJ quantum circuit tells us whether f is constant or balanced with only one question.

For the classical case, two questions need to be asked.

This problem is artificial, but was the first problem discovered with a "quantum advantage."

Other problems (which are NOT artificial!) have been shown to have a quantum advantage as well.

Conclusions and takeaways

We measure 0 if and only if f is constant.

We measure 1 if and only if f is balanced.

Running the DJ quantum circuit tells us whether f is constant or balanced with only one question.

For the classical case, two questions need to be asked.

Conclusions and takeaways

We measure 0 if and only if f is constant.

We measure 1 if and only if f is balanced.

Running the DJ quantum circuit tells us whether f is constant or balanced with only one question.

For the classical case, two questions need to be asked.

This problem is artificial, but was the first problem discovered with a “quantum advantage.”

Conclusions and takeaways

We measure 0 if and only if f is constant.

We measure 1 if and only if f is balanced.

Running the DJ quantum circuit tells us whether f is constant or balanced with only one question.

For the classical case, two questions need to be asked.

This problem is artificial, but was the first problem discovered with a “quantum advantage.”

Other problems (which are NOT artificial!) have been shown to have a quantum advantage as well.