5. The Algorithms of Deutsch and Simon

Myeonghyeon Kim

June 10th, 2025

Outline

Introduction

Introduction

- Can quantum computers solve certain problems faster than classical ones?

Introduction

- Can quantum computers solve certain problems faster than classical ones?
- If so, for which types of problems? How much faster?

Introduction

- Can quantum computers solve certain problems faster than classical ones?
- If so, for which types of problems? How much faster?
- We will explore two foundational quantum algorithms:

Introduction

- Can quantum computers solve certain problems faster than classical ones?
- If so, for which types of problems? How much faster?
- We will explore two foundational quantum algorithms:
  - **Deutsch–Jozsa Algorithm**

Introduction

- Can quantum computers solve certain problems faster than classical ones?
- If so, for which types of problems? How much faster?
- We will explore two foundational quantum algorithms:
  - **Deutsch–Jozsa Algorithm**
  - **Simon's Algorithm**

Introduction

- Can quantum computers solve certain problems faster than classical ones?
- If so, for which types of problems? How much faster?
- We will explore two foundational quantum algorithms:
  - **Deutsch–Jozsa Algorithm**
  - **Simon's Algorithm**
- Goal: Understand quantum speedup through concrete examples.

Deutsch-Jozsa Algorithm

The Problem Setting

Black box (oracle): A system (function/operator) treated as unknown internally, known only through inputs and outputs.

Function type classification:
For a function $f : \{0,1\}^n \to \{0,1\}$, we classify it as:

- Constant: All inputs map to the same output

$$|f^{-1}(0)| = 2^n \quad \text{or} \quad |f^{-1}(1)| = 2^n$$

- Balanced: Half the inputs map to 0, the other half to 1

$$|f^{-1}(0)| = |f^{-1}(1)| = 2^{n-1}$$

Examples: Case $n = 1$

All possible functions when $n = 1$

x	0	1	type
f_1	0	0	constant
f_2	1	1	constant
f_3	0	1	balanced
f_4	1	0	balanced

$\Rightarrow$ Only 4 possible functions, 2 constant and 2 balanced

Examples: Case $n = 2$

Representative functions when $n = 2$

x	00	01	10	11	type
f_1	0	0	0	0	constant
f_2	1	1	1	1	constant
f_3	1	0	0	0	neither
f_4	1	1	0	0	balanced
f_5	1	1	1	0	neither
$\vdots$				$\vdots$	
f_{16}	0	1	1	1	neither

Out of $2^{|\{0,1\}^2|} = 16$ total functions, only:

- 2 are constant
- 6 are balanced

Deutsch–Jozsa Problem

Problem Setup:

Let $f : \{0,1\}^n \to \{0,1\}$ be a black-box function. Assume f is either Constant or Balanced.

Deutsch–Jozsa Problem: How many queries are needed to determine whether f is constant or balanced?

Remark: The Deutsch problem is a special case where $n = 1$.

Classical Solution (Lower Bound)

Exercise 5.3.2 Show that every deterministic classical algorithm needs at least $2^{n-1}+1$ queries to solve the problem.

Idea: There exist cases where 2^{n-1} queries are not enough to distinguish constant from balanced.

x	x_1	$\cdots$	$x_{2^{n-1}}$	$x_{2^{n-1}+1}$	$\cdots$	x_{2^n}	type
f_1	0	$\cdots$	0	0	$\cdots$	0	constant
f_2	0	$\cdots$	0	1	$\cdots$	1	balanced

Classical Solution (Argument)

Even if $f(x_1), \ldots, f(x_{2^{n-1}}) = 0$, we cannot tell if f is constant or balanced.

We need one more query. Query $x_{2^{n-1}+1}$:

- If $f(x_i) \neq f(x_j)$ for some i, j, then f is balanced
- If all queried outputs are equal:

$$|f^{-1}(0)| \geq 2^{n-1} + 1 \quad \text{or} \quad |f^{-1}(1)| \geq 2^{n-1} + 1$$

Since the only valid sizes for constant/balanced are $0, 2^{n-1}, 2^n$, we conclude:

$$|f^{-1}(0)| = 2^n \quad \text{or} \quad |f^{-1}(1)| = 2^n$$

$\Rightarrow f$ is constant.

Conclusion: Minimum required queries $= 2^{n-1} + 1$

What is Query (Oracle) Complexity?

Classical Oracle

- Black-box returns $f(x)$ for input x
- Internal logic is unknown
- Each query reveals info about one input
- Used in deterministic or randomized algorithms

Quantum Oracle

- Unitary operator U_f encoding $f : \{0,1\}^n \to \{0,1\}^k$
- Often defined as:

$$U_f : \mathbb{H}_n \otimes \mathbb{H}_k \to \mathbb{H}_n \otimes \mathbb{H}_k,$$
$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$$

- Can be queried with superpositions of inputs

Query complexity = the number of oracle calls needed to solve the problem

Quantum Version of the Problem

Quantum Oracle Definition:

- Given $f : \{0,1\}^n \to \{0,1\}$, define the unitary operator:

$$U_f : \mathbb{H}_n \otimes \mathbb{H}_1 \to \mathbb{H}_n \otimes \mathbb{H}_1$$

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$

Deutsch–Jozsa Operator:

- Define

$$Q_f^{DJ} := (H^{\otimes n} \otimes I) \cdot U_f \cdot (H^{\otimes n} \otimes H)$$

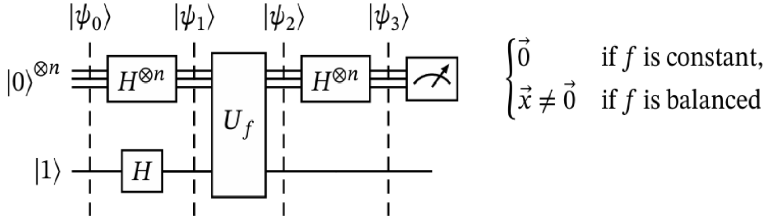

Figure 5.3.1. The quantum circuit $Q_{DJ}(n, U_f)$ that solves the Deutsch-Jozsa problem.

Hadamard Operator: Key Properties

Recall: Properties of the Hadamard Operator H

- **Action on Basis:**

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} =: |x_+\rangle, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} =: |x_-\rangle$$

- **Self-inverse:** $H^2 = I$

$$\begin{aligned} H(H|0\rangle) &= H\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) = \frac{1}{\sqrt{2}}(H|0\rangle + H|1\rangle) \\ &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) = |0\rangle \end{aligned}$$

Similarly,

$$H(H|1\rangle) = |1\rangle \quad \Rightarrow \quad H^2 = I$$

Therefore,

$$H^{\otimes n} H^{\otimes n} = (H^2)^{\otimes n} = I^{\otimes n}$$

Quantum Deutsch–Jozsa: Main Idea

Theorem 5.3.7 Let $f : \{0,1\}^n \to \{0,1\}$ be either constant or balanced. Then:

Final measurement of the first register yields $\lvert 0\rangle^{\otimes n} \Leftrightarrow f$ is constant

- **Step 1:** Initial state

$$\lvert\psi_0\rangle = \lvert 0\rangle^{\otimes n} \lvert 1\rangle$$

- **Step 2:** Apply $H^{\otimes n+1}$

$$\lvert\psi_1\rangle = \left(\frac{\lvert 0\rangle + \lvert 1\rangle}{\sqrt{2}}\right)^{\otimes n} \otimes \frac{\lvert 0\rangle - \lvert 1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} \lvert y\rangle \otimes \lvert x_-\rangle$$

Step 3: Apply Oracle U_f

- **Apply** U_f:

$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} U_f|y\rangle|x_-\rangle$$

- **For each** $y \in \{0,1\}^n$:

$$\begin{aligned} U_f|y\rangle|x_-\rangle &= \frac{1}{\sqrt{2}}\left(U_f|y\rangle|0\rangle - U_f|y\rangle|1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|y\rangle|f(y)\rangle - |y\rangle|f(y)\oplus 1\rangle\right) \\ &= |y\rangle\left(\frac{|f(y)\rangle - |f(y)\oplus 1\rangle}{\sqrt{2}}\right) = (-1)^{f(y)}|y\rangle|x_-\rangle \end{aligned}$$

- **Therefore:**

$$|\psi_2\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{f(y)}|y\rangle\right) \otimes |x_-\rangle$$

Step 4: Final Hadamard and Measurement

- **Apply $H^{\otimes n} \otimes I$ to ψ_2:**

$$|\psi_3\rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{f(y)} H^{\otimes n} |y\rangle \right) \otimes |x_-\rangle$$

- **Recall single-qubit transform:**

$$H|k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^k |1\rangle) = \frac{1}{\sqrt{2}} \sum_{z \in \{0,1\}} (-1)^{k \cdot z} |z\rangle$$

- This allows us to expand $H^{\otimes n}|y\rangle$ as a sum over z with $(-1)^{y \cdot z}$ weights

Quantum version and its solution

This implies that for each $y = (y_0, \ldots, y_{n-1}) \in \{0,1\}^n$,

$$
\begin{aligned}
H^{\otimes n}|y\rangle &= \frac{1}{\sqrt{2^n}} \left(\sum_{z_0 \in \{0,1\}} (-1)^{y_0 z_0} |z_0\rangle \right) \otimes \cdots \otimes \left(\sum_{z_{n-1} \in \{0,1\}} (-1)^{y_{n-1} z_{n-1}} |z_{n-1}\rangle \right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{(z_0, \ldots, z_{n-1}) \in \{0,1\}^n} ((-1)^{y_0 z_0} |z_0\rangle) \otimes \cdots \otimes ((-1)^{y_{n-1} z_{n-1}} |z_{n-1}\rangle) \\
&= \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{y \cdot z} |z\rangle.
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
|\psi_3\rangle &= \left(\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{f(y)} \left[\frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{y \cdot z} |z\rangle \right] \right) \otimes |x_-\rangle \\
&= \left(\sum_{z \in \{0,1\}^n} \left[\frac{1}{2^n} \sum_{y \in \{0,1\}^n} (-1)^{f(y)} (-1)^{y \cdot z} \right] |z\rangle \right) \otimes |x_-\rangle.
\end{aligned}
$$

Quantum version and its solution

We now measure the first register of $|\psi_3\rangle$ in the computational basis $\{|z\rangle\}_{z\in\{0,1\}^n}$, which gives us a state $|z\rangle$ with probability

$$\left|\frac{1}{2^n}\sum_{y\in\{0,1\}^n}(-1)^{f(y)}(-1)^{y\cdot z}\right|^2.$$

Since the coefficient of $|0\rangle^{\otimes n}$ in the first register of $|\psi_3\rangle$ is

$$\frac{1}{2^n}\sum_{y\in\{0,1\}^n}(-1)^{f(y)} = \frac{1}{2^n}\left(\sum_{y\in f^{-1}(0)}(-1)^{f(y)} + \sum_{y\in f^{-1}(1)}(-1)^{f(y)}\right)$$

$$= \frac{1}{2^n}\left(|f^{-1}(0)| - |f^{-1}(1)|\right) = \begin{cases}\pm 1 & \text{if } f \text{ is constant,}\\ 0 & \text{if } f \text{ is balanced,}\end{cases}$$

we can see that the probability for a state $|0\rangle^{\otimes n}$ to be measured is

$$\left|\frac{1}{2^n}\sum_{y\in\{0,1\}^n}(-1)^{f(y)}\right|^2 = \begin{cases}1 & \text{if } f \text{ is constant,}\\ 0 & \text{if } f \text{ is balanced.}\end{cases}$$

Simon's Algorithm

Simon's Problem: Definition

Definition 3.1
Let $f : \{0,1\}^n \to \{0,1\}^n$ be a black-box function. Assume there exists a nonzero vector $s \in \{0,1\}^n$ such that:

$$f(x) = f(y) \quad \text{if and only if} \quad x = y \quad \text{or} \quad x = y \oplus s$$

Here, $\oplus$ denotes bitwise XOR.

The string s is called the hidden string.

Goal: Determine the hidden string s using as few queries to f as possible.

Classical Lower Bound

Exercise 5.4.2 Let A be a classical deterministic algorithm solving Simon's problem. Show: A must make at least $2^{n-1}+1$ queries in the worst case.

Solution Idea: Show that 2^{n-1} queries are not enough to determine s, but $2^{n-1}+1$ queries are sufficient.

Let $\{x_1 = \vec{0}, x_2, \ldots, x_{2^n}\} \subset \{0,1\}^n$. For any $s \neq 0$, the domain can be partitioned as:

$$\{0,1\}^n = \bigsqcup_{i=1}^{2^{n-1}} \{x_i, x_i \oplus s\}$$

Choose two different nonzero strings $s_1 \neq s_2$, define:

$$g(x_i) = g(x_i \oplus s_1) = x_i, \quad h(x_i) = h(x_i \oplus s_2) = x_i$$

for $i = 1, \ldots, 2^{n-1}$

Classical Lower Bound (continued)

Observation: By the identity

$$\{0,1\}^n = \bigsqcup_{i=1}^{2^{n-1}} \{x_i, x_i \oplus s\}$$

the functions g and h are well-defined and satisfy Simon's condition.

So even if we observe:

$$f(x_i) = x_i \quad \text{for } i = 1, \ldots, 2^{n-1}$$

we cannot distinguish between $f = g$ or $f = h \Rightarrow$ cannot determine the hidden string s yet.

Classical Lower Bound (continued)

Now suppose: We query f at $2^{n-1}+1$ distinct inputs.

Since

$$f(x) = f(y) \quad \Leftrightarrow \quad x = y \quad \text{or} \quad x = y \oplus s$$

f is a **2-to-1** function.

Therefore:

$$|\mathrm{Im}(f)| = \frac{|\{0,1\}^n|}{2} = 2^{n-1}$$

Querying more than 2^{n-1} inputs guarantees a collision $\Rightarrow$ allows recovery of the hidden string s.

Classical Solution (Final Argument)

We can write the domain as:

$$\{0,1\}^n = \bigsqcup_{b \in \mathrm{Im}(f)} f^{-1}(b), \quad \text{with } |\mathrm{Im}(f)| = 2^{n-1}$$

By the pigeonhole principle, querying $2^{n-1}+1$ inputs yields some $j < k$ such that:

$$f(x_j) = f(x_k)$$

Using the Simon condition:

$$x_j = x_k \oplus s \quad \Rightarrow \quad x_k \oplus x_j = s$$

Conclusion: With $2^{n-1}+1$ classical queries, we can recover the hidden string s.

Quantum Oracle and Operator

Definition 3.2
Let $f : \{0,1\}^n \to \{0,1\}^n$ be a black-box function.

1. **Quantum Oracle:**

$$U_f : \mathbb{H}_n \otimes \mathbb{H}_n \to \mathbb{H}_n \otimes \mathbb{H}_n, \quad U_f|x\rangle|y\rangle = |x\rangle|f(x) \oplus y\rangle$$

2. **Simon's Operator:**

$$Q_f^S := (H^{\otimes n} \otimes I^{\otimes n}) \cdot U_f \cdot (H^{\otimes n} \otimes I^{\otimes n})$$

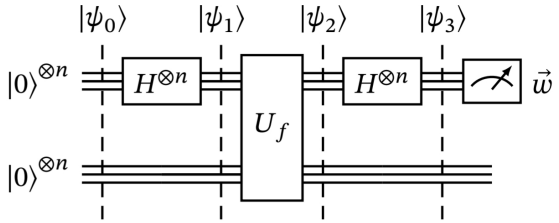

Figure 5.4.1. The quantum circuit $Q_{\text{Simon}}(n, U_f)$ used in Simon's algorithm.

Simon's Algorithm: Overview

Main Ideas

1. Run the quantum subroutine $n-1$ times to get $w_1, \ldots, w_{n-1}$ such that
$$w_k \cdot s = 0$$
2. If $w_1, \ldots, w_{n-1}$ are linearly independent, solve the system:
$$\text{Solve } W^T x = 0^n \quad \Rightarrow \quad x = s$$
3. If not, the algorithm fails to determine s.

Step 1: Initialize
$$|\psi_0\rangle = |0\rangle^{\otimes n}|0\rangle^{\otimes n}$$

Step 2: Apply Hadamard to first register:
$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} |y\rangle \otimes |0\rangle^{\otimes n}$$

Simon's Algorithm: Continued

Step 3: Apply the oracle

$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} |y\rangle \otimes |f(y)\rangle$$

Step 4: Apply Hadamard again to first register:

$$|\psi_3\rangle = (H^{\otimes n} \otimes I^{\otimes n})|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} H^{\otimes n}|y\rangle \otimes |f(y)\rangle$$

Recall:

$$H^{\otimes n}|y\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{y\cdot z}|z\rangle$$

So:

$$|\psi_3\rangle = \sum_{z\in\{0,1\}^n} \left[\frac{1}{2^n} \sum_{y\in\{0,1\}^n} (-1)^{y\cdot z}|f(y)\rangle \right] \otimes |z\rangle$$

Simon's Algorithm

By interchanging the order of summation, we have

$$|\psi_3\rangle = \sum_{z\in\{0,1\}^n} |z\rangle \left(\left[\frac{1}{2^n} \sum_{y\in\{0,1\}^n} (-1)^{y\cdot z} \right] \otimes |f(y)\rangle \right).$$

Finally, we measure the first register. Note that there exists a string x_k, such that $f(x_k) = f(x_k \oplus s) = k$ for each $k \in Imf$. Thus, the probability of measuring $|j\rangle$ is

$$|\langle j|\psi_3\rangle|^2 = \left\| \frac{1}{2^n} \sum_{y\in\{0,1\}^n} (-1)^{y\cdot j} |f(y)\rangle \right\|^2 = \left\| \frac{1}{2^n} \sum_{k\in Imf} \left((-1)^{x_k\cdot j} + (-1)^{(x_k\oplus s)\cdot j} \right) |k\rangle \right\|^2$$

$$= \left\| \frac{1}{2^n} \sum_{k\in Imf} \left((-1)^{x_k\cdot j} + (-1)^{x_k\cdot j \,\oplus\, s\cdot j} \right) |k\rangle \right\|^2 = \left\| \frac{1}{2^n} \sum_{k\in Imf} (-1)^{x_k\cdot j} \left(1 + (-1)^{s\cdot j} \right) |k\rangle \right\|^2$$

$$= \frac{1}{2^{2n}} \sum_{k\in Imf} (-1)^{2x_k\cdot j} (1 + (-1)^{s\cdot j})^2 = \begin{cases} 2^2|Imf|/2^{2n} = 1/2^{n-1} & \text{if } s\cdot j = 0, \\ 0 & \text{if } s\cdot j = 1. \end{cases}$$

Thus, our measured j satisfies $s \cdot j = 0$, i.e., $j \in s^{\perp}$.

Simon's Algorithm

- **Step 5:** Repeat the above steps $n-1$ times to get bitstrings $w_1, \ldots, w_{n-1}$. Suppose that $w_1, \ldots, w_{n-1}$ are linearly independent(otherwise, the algorithm terminates without recovering s). Then, $W := (w_1 | \ldots | w_{n-1}) \in \mathbb{Z}_2^{n \times (n-1)}$ is of rank $n-1$. Applying the rank theorem to the map

$$W^T : \mathbb{Z}_2^n \to \mathbb{Z}_2^{n-1}, \quad x \mapsto W^T x,$$

we have
$n = \dim \mathbb{Z}_2^n = \dim(\ker W^T) + \mathrm{rank}(W^T) = \dim(\ker W^T) + (n-1)$.
Since $w_k \cdot s = 0$ for all $k = 1, \ldots, n-1$, we have $s \in \ker W^T$.
Since $\ker W^T$ is a subspace of $\mathbb{Z}_2^n$ of dimension 1, s is the only non-zero vector in $\ker W^T$,i.e., $\ker W^T = \{0, s\}$. Hence, we can uniquely determine the hidden string s.

Success probability of Simon's Algorithm

Proposition 5.4.13 In each execution, the probability that $w_1, \ldots, w_{n-1}$ are linearly independent is at least

$$\prod_{k=1}^{\infty}\left(1-\frac{1}{2^k}\right) \geq \frac{1}{4}.$$

(Solution) Suppose we already have $k-1$ linearly independent vectors $w_1, \ldots, w_{k-1}$ in $\mathbb{Z}_2^n$. Then, $\mathbf{span}\{w_1, \ldots, w_{k-1}\}$ is a $(k-1)$-dimensional subspace of $s^{\perp}$ with $\dim s^{\perp} = n-1$(over $\mathbb{Z}_2$), which implies that the probability that the next vector w_k is linearly independent from $w_1, \ldots, w_{k-1}$ is

$$\frac{|s^{\perp}| - |\mathbf{span}\{w_1, \ldots, w_{k-1}\}|}{|s^{\perp}|} = \frac{2^{n-1}-2^{k-1}}{2^{n-1}} = 1-\frac{1}{2^{n-k}}.$$

Then, the probability that $w_1, \ldots, w_{n-1}$ are linearly independent is

$$\prod_{k=1}^{n-1}\left(1-\frac{1}{2^{n-k}}\right) = \prod_{k=1}^{n-1}\left(1-\frac{1}{2^k}\right) \geq \prod_{k=1}^{\infty}\left(1-\frac{1}{2^k}\right) \geq \frac{1}{4} \text{ (by Lem5.4.11, p.217)}$$

Success probability of Simon's Algorithm

Remark
From the above proposition, we can see that after repeating the entire process $4m$ times, the probability of not finding a linearly independent set during one of the iterations is

$$\left(1 - \frac{1}{4}\right)^{4m} < e^{-m}.$$

Hence, we conclude that for any constant $\epsilon > 0$, Simon's algorithm can solve the problem we are considering with error probability at most ϵ using $O(n)$ queries to the black-box(quantum oracle).

Quantum vs Classical: Is It Fair?

- You might think it's unfair that a classical deterministic algorithm solves the problem with probability 1,
- while Simon's algorithm succeeds with a probability of about $1/4$.
- So, a natural question is:
  Can some classical randomized algorithm be faster than Simon's algorithm?

Classical probabilistic algorithm

Theorem 5.4.3 Any classical probabilistic algorithm that solves Simon's problem with probability at least $\frac{1}{4}$ must make $\Omega(2^{n/2})$ queries to the black-box for f.

Simon's algorithm achieves an exponential speedup
in oracle complexity over any classical probabilistic algorithm.

Classical probabilistic algorithm

(Solution) Let f be a black box function that satisfies the assumption of Simon's problem with a hidden string $s \in \{0,1\}^n - \{0^n\}$. We aim to analyze how much information we can get from the $(k+1)$th query when we have already queried f k-times and there is no such pair of inputs x_i, x_j such that $f(x_i) = f(x_j)$ because if there is such a pair, we can determine the hidden string $s = x_i \oplus x_j$, as seen in the classical solution. Then, the queried $f(x_i)$'s are distinct and s is none of $\binom{k}{2}$ values $x_i \oplus x_j (1 \leq i < j \leq k)$.

Classical probabilistic algorithm

The probability that the next query will succeed is at most

$$\frac{|\{a \in \{0,1\}^n \setminus \{0^n\} \mid \exists\, i \in [k],\ a = x_{k+1} \oplus x_i\}|}{|\{a \in \{0,1\}^n \setminus \{0^n\} \mid a \neq x_i \oplus x_j \text{ for all } 1 \leq i < j \leq k\}|} \leq \frac{k}{2^n - 1 - \binom{k}{2}}$$

Taking the sum over all $k = 1, \ldots, m$, we can see that the probability of finding a hidden string s after m queries is at most

$$\sum_{k=1}^{m} \frac{k}{2^n - 1 - (k-1)(k-2)/2} \leq \sum_{k=1}^{m} \frac{2k}{2^{n+1} - k^2}$$
$$\leq \sum_{k=1}^{m} \frac{2m}{2^{n+1} - m^2} = \frac{2m^2}{2^{n+1} - m^2}.$$

If this quantity is to be at least $\frac{1}{4}$, then

$$\frac{2m^2}{2^{n+1} - m^2} \geq \frac{1}{4} \quad \Longrightarrow m \geq \sqrt{\frac{2}{9} 2^n} \quad \Longrightarrow m = \Omega(2^{n/2}).$$