Complexity of recognizing Dyck languages of bounded height with quantum query algorithms.

Maxime CAUTRÈS

Faculty of Computing
University of Latvia

31/08/2022

Sommaire

Classical and quantum computers are both made with simple components.

a
b
c

Figure: A Boolean circuit (Full adder).

$|a\rangle$
$|b\rangle$
$|c\rangle$

Figure: A Quantum circuit.

Classical and quantum computers are both made with simple components.

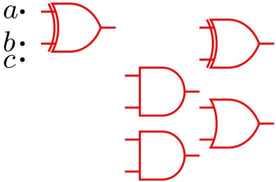


Figure: A Boolean circuit (Full adder).

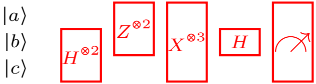


Figure: A Quantum circuit.

Classical and quantum computers are both made with simple components.

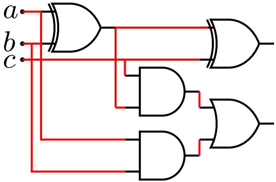


Figure: A Boolean circuit (Full adder).

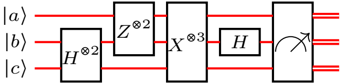


Figure: A Quantum circuit.

Classical and quantum computers are both made with simple components.

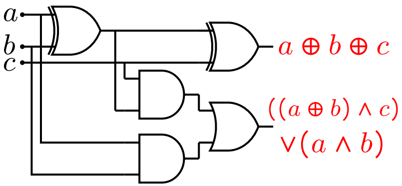


Figure: A Boolean circuit (Full adder).

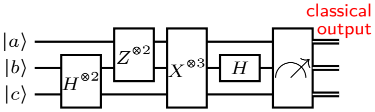


Figure: A Quantum circuit.

Interacting with qubits is more complexe.

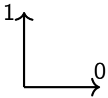


Figure: A classical bit

Interacting with qubits is more complexe.

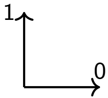


Figure: A classical bit

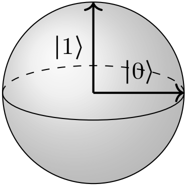


Figure: A quantum bit.

Interacting with qubits is more complexe.

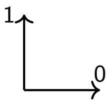


Figure: A classical bit

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Figure: Truth table on 2 bits.

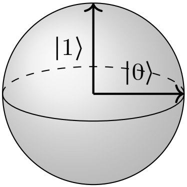


Figure: A quantum bit.

Interacting with qubits is more complexe.

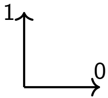


Figure: A classical bit

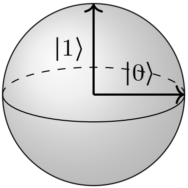


Figure: A quantum bit.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Figure: Truth table on 2 bits.

$$H^{\otimes 2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Figure: Unitary matrix on 2 qubits.

Quantum query algorithm is just a quantum circuit.

$$x = \underbrace{100101\ldots01011}_{n}$$

Figure: Structure of a quantum query algorithm.

Quantum query algorithm is just a quantum circuit.

$$x = \underbrace{100101\ldots01011}_{n}$$

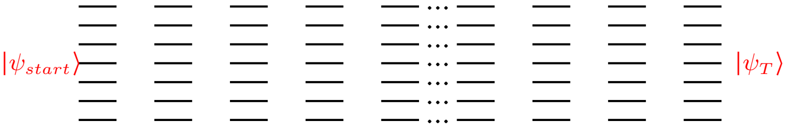


Figure: Structure of a quantum query algorithm.

Quantum query algorithm is just a quantum circuit.

$$x = \underbrace{100101\ldots01011}_{n}$$

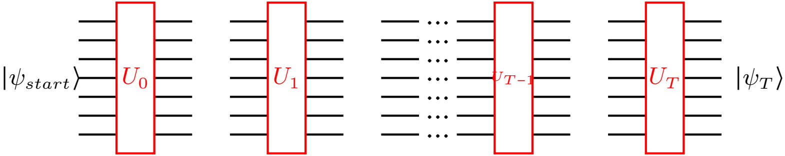


Figure: Structure of a quantum query algorithm.

Quantum query algorithm is just a quantum circuit.

$$x = \underbrace{100101\ldots01011}_{n}$$

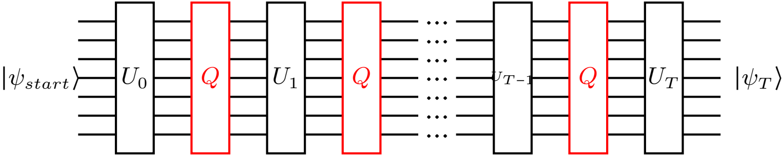


Figure: Structure of a quantum query algorithm.

Quantum query algorithm is just a quantum circuit.

$$x = \underbrace{100101\ldots01011}_{n}$$

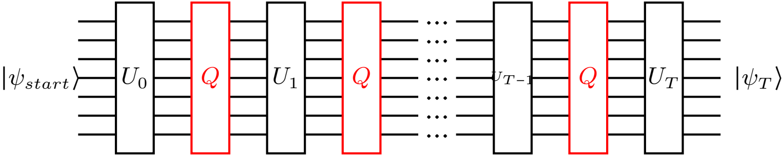


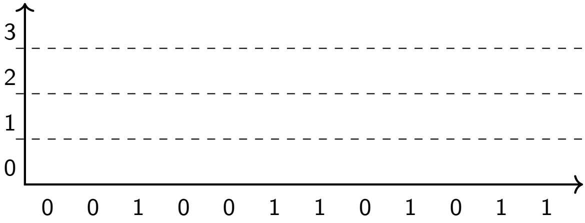
Figure: Structure of a quantum query algorithm.

$$Q(f)$$

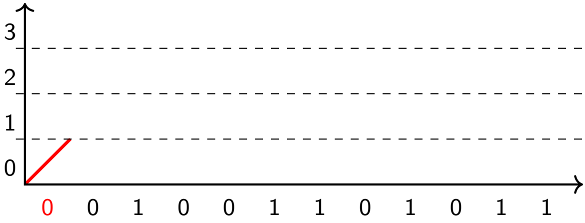
Dyck words of bounded height are a natural restriction of Dyck words.

0 0 1 0 0 1 1 0 1 0 1 1

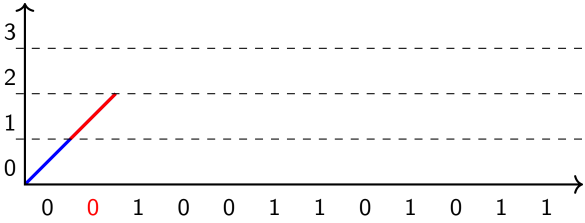
Dyck words of bounded height are a natural restriction of Dyck words.



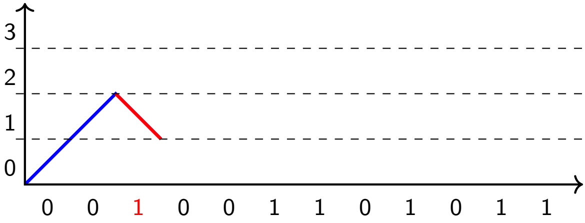
Dyck words of bounded height are a natural restriction of Dyck words.



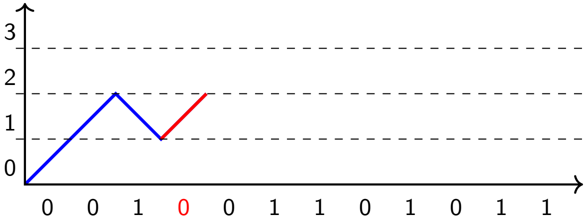
Dyck words of bounded height are a natural restriction of Dyck words.



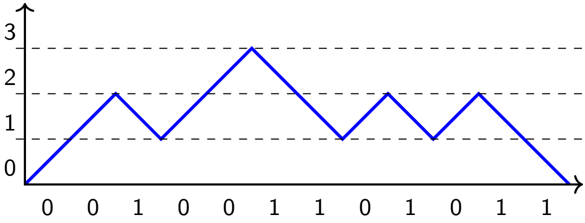
Dyck words of bounded height are a natural restriction of Dyck words.



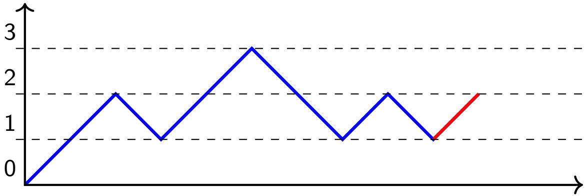
Dyck words of bounded height are a natural restriction of Dyck words.



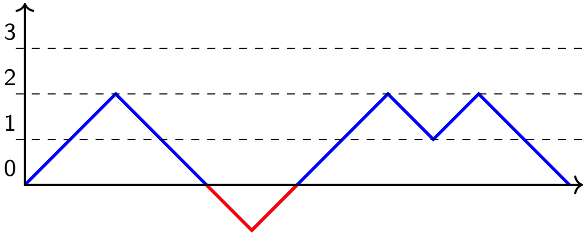
Dyck words of bounded height are a natural restriction of Dyck words.



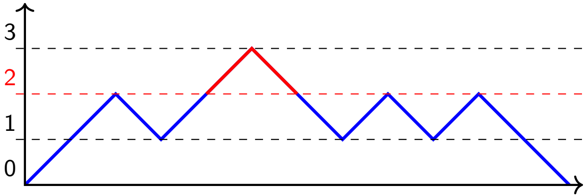
Dyck words of bounded height are a natural restriction of Dyck words.



Dyck words of bounded height are a natural restriction of Dyck words.



Dyck words of bounded height are a natural restriction of Dyck words.



$$\textsc{Dyck}_k$$

A general result that help but not close the $Q(\textsc{Dyck}_k)$ problem.

The Trichotomy theorem:(Aaronson, Grier and Schaeffer [**?**, 2019])

$$\text{Star Free Languages} \implies \tilde{\Theta}\left(\sqrt{n}\right)$$

A general result that help but not close the $Q(\textsc{Dyck}_k)$ problem.

The Trichotomy theorem:(Aaronson, Grier and Schaeffer [**?**, 2019])

$$\text{Star Free Languages} \implies \tilde{\Theta}\left(\sqrt{n}\right)$$

Application:

$$\textsc{Dyck}_k \in \text{Star free languages}$$

A general result that help but not close the $Q(\text{DYCK}_k)$ problem.

The Trichotomy theorem:(Aaronson, Grier and Schaeffer [?, 2019])

$$\text{Star Free Languages} \implies \tilde{\Theta}\left(\sqrt{n}\right)$$

Application:

$$\text{DYCK}_k \in \text{Star free languages}$$

Implication:

$$Q(\text{DYCK}_{k,n}) = \Theta\left(\sqrt{n}\log_2(n)^{p(k)}\right)$$

First step, one try to have a good upper bounds.

- **Algorithms:**

Require: $n \geq 0$ and $k \geq 1$
Ensure: $|x| = n$
$x \leftarrow 1^k x 0^k$
$v \leftarrow \text{FINDANY}_{k+1}(0, n+2*k-1, \{1,-1\})$
return v $= \text{NULL}$

Figure: Ambainis' algorithm (small piece).

First step, one try to have a good upper bounds.

- **Algorithms:**

Require: $n \geq 0$ and $k \geq 1$
Ensure: $|x| = n$
$x \leftarrow 1^k x 0^k$
$v \leftarrow \text{FINDANY}_{k+1}(0, n + 2 * k - 1, \{1, -1\})$
return v $= \text{NULL}$

Figure: Ambainis' algorithm (small piece).

$$O\left(\sqrt{n}(\log_2(n))^{0.5k}\right)$$

First step, one try to have a good upper bounds.

- **Algorithms:**

Require: $n \geq 0$ and $k \geq 1$
Ensure: $|x| = n$
$x \leftarrow 1^k x 0^k$
$v \leftarrow \text{FINDANY}_{k+1}(0, n + 2 * k - 1, \{1, -1\})$
return v $= \text{NULL}$

Figure: Ambainis' algorithm (small piece).

$$O\left(\sqrt{n}(\log_2(n))^{0.5k}\right)$$

- **Reductions to:**

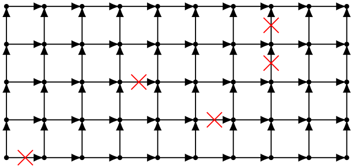


Figure: A reduction to 2D directed grid connectivity.

First step, one try to have a good upper bounds.

- **Algorithms:**

Require: $n \geq 0$ and $k \geq 1$
Ensure: $|x| = n$
$x \leftarrow 1^k x 0^k$
$v \leftarrow \text{FINDANY}_{k+1}(0, n + 2 * k - 1, \{1, -1\})$
return v $= \text{NULL}$

Figure: Ambainis' algorithm (small piece).

$$O\left(\sqrt{n}(\log_2(n))^{0.5k}\right)$$

- **Reductions to:**

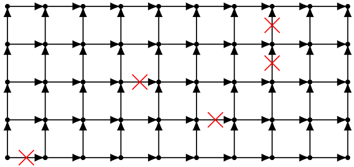


Figure: A reduction to 2D directed grid connectivity.

$$O\left(\sqrt{n}(\log_2(n))^{0.5(k-1)}\right)$$

Second step, one try to prove the optimality with a matching lower bound.

- **Adversary methods:**

No result yet

Second step, one try to prove the optimality with a matching lower bound.

- **Adversary methods:**

No result yet

- **Reduction from:**

$$\mathrm{Ex}_{2m}^{m|m+1}(x) = 0 \Leftrightarrow |x|_0 - |x|_1 = 2$$
$$\mathrm{Ex}_{2m}^{m|m+1}(x) = 1 \Leftrightarrow |x|_0 - |x|_1 = 0$$

$$\Omega\left(\sqrt{n}c^k\right)$$

A natural goal is to made the bounds match.

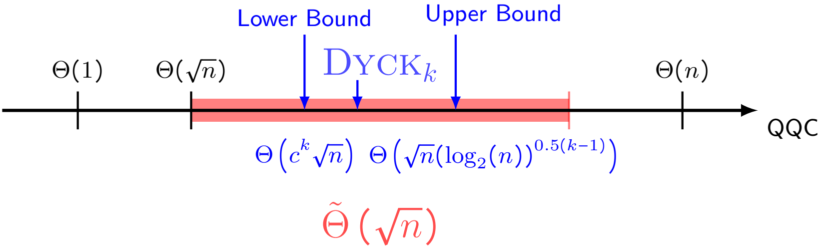


Figure: Representation of the different bounds.

Sommaire

1. Introduction
2. The progress to reduce the $\mathrm{DYCK}_{k,n}$ QQC
   - Why does the problem is not only a grover search
   - Original algorithm and small revisions
   - A new algorithm for k=2
3. New idea to get better quantum query complexity bounds

Every k is not as simple as 1.

- $k = 1$:

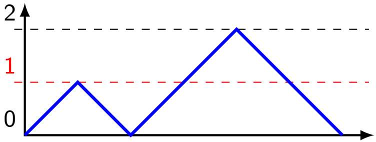


Figure: A dyck word of height 2.

Every k is not as simple as 1.

- $k = 1$:

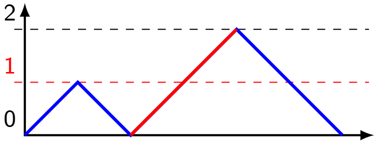


Figure: A dyck word of height 2.

$$O\left(\sqrt{n}\right)$$

Every k is not as simple as 1.

- $k = 1$:

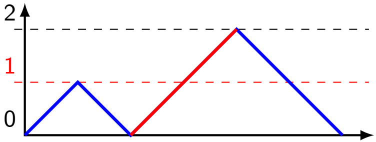


Figure: A dyck word of height 2.

$$O\left(\sqrt{n}\right)$$

- $k = 2$:

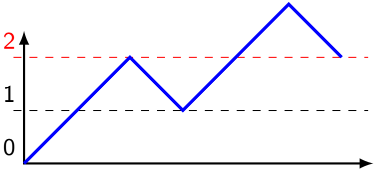


Figure: A substring of height 3.

Every k is not as simple as 1.

- $k = 1$:

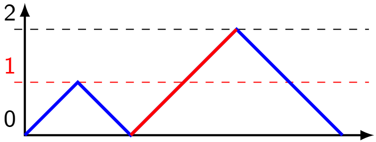


Figure: A dyck word of height 2.

$$O\left(\sqrt{n}\right)$$

- $k = 2$:

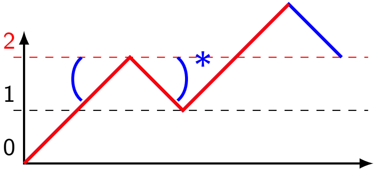


Figure: A substring of height 3.

Every k is not as simple as 1.

- $k = 1$:

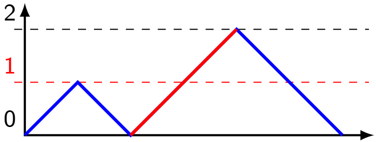


Figure: A dyck word of height 2.

$$O\left(\sqrt{n}\right)$$

- $k = 2$:

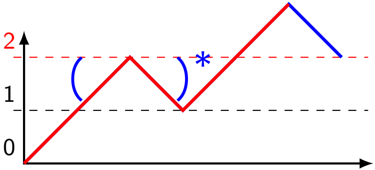


Figure: A substring of height 3.

$$O\left(\sqrt{n\log_2(n)}\right)$$

Small definition and intuition

- $\pm k$ **strings:**

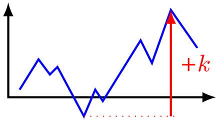


Figure: Representation of a $+k$ string.

Small definition and intuition

- $\pm k$ **strings:**

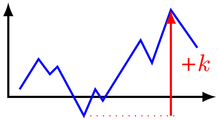


Figure: Representation of a $+k$ string.

- **Minimal** $\pm k$ **strings:**

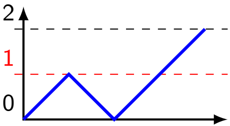


Figure: A non-minimal $+2$ string.

- **Minimal decomposition**

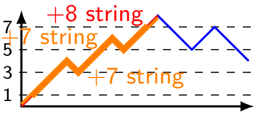


Figure: A $+3$ string decomposition.

A REFAIRE PLUS PETIT

Small definition and intuition

- **$\pm k$ strings:**

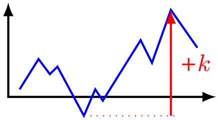


Figure: Representation of a $+k$ string.

- **Minimal $\pm k$ strings:**

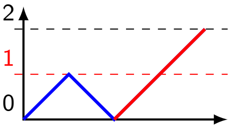


Figure: A non-minimal $+2$ string.

Small definition and intuition

- **$\pm k$ strings:**

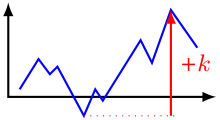


Figure: Representation of a $+k$ string.

- **Minimal $\pm k$ strings:**

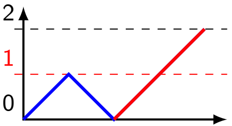


Figure: A non-minimal $+2$ string.

- **Minimal decomposition**

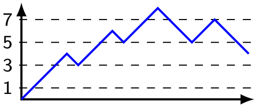


Figure: A $+3$ string decomposition.

A REFAIRE PLUS PETIT

Small definition and intuition

- **$\pm k$ strings:**

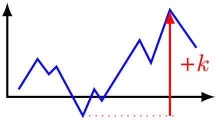


Figure: Representation of a $+k$ string.

- **Minimal $\pm k$ strings:**

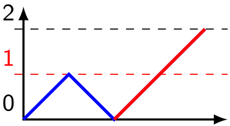


Figure: A non-minimal $+2$ string.

- **Minimal decomposition**

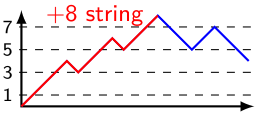


Figure: A $+3$ string decomposition.

A REFAIRE PLUS PETIT

Small definition and intuition

- **$\pm k$ strings:**

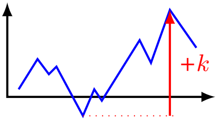


Figure: Representation of a $+k$ string.

- **Minimal $\pm k$ strings:**

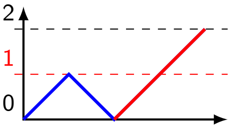


Figure: A non-minimal $+2$ string.

- **Minimal decomposition**

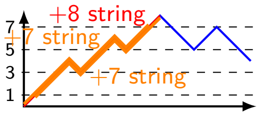


Figure: A +3 string decomposition.

A REFAIRE PLUS PETIT

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

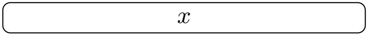


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}(\sqrt{n})$

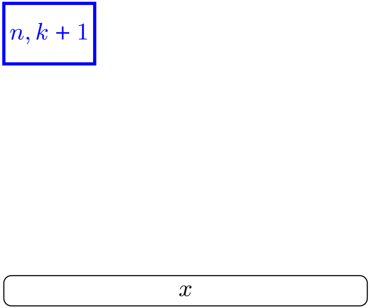


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

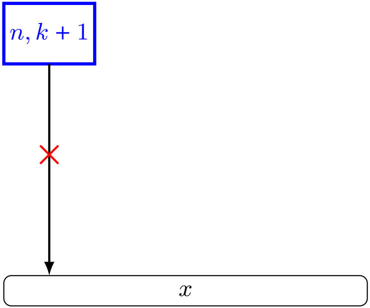


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

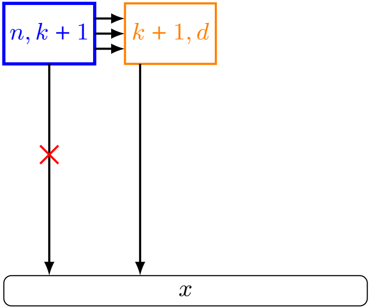


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

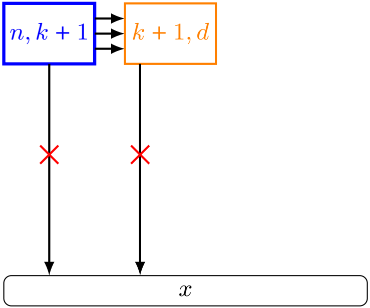


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

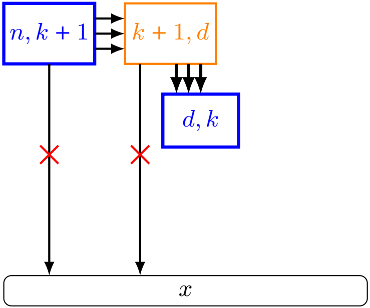


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

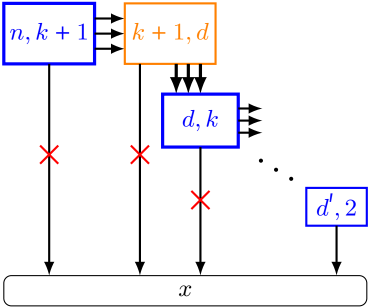


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

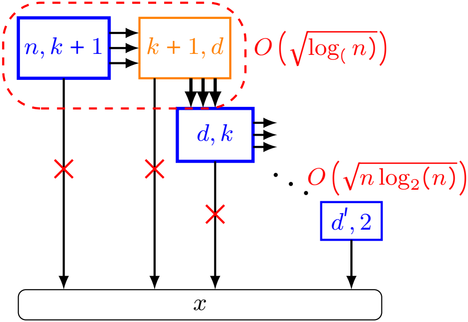


Figure: Schema of the idea of the original algorithm.

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

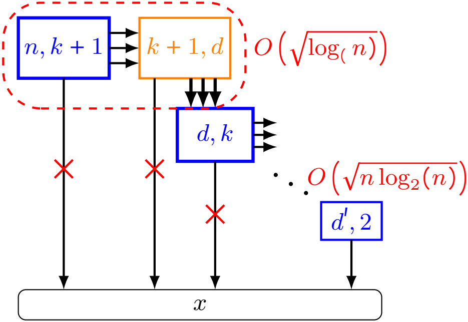


Figure: Schema of the idea of the original algorithm.

- **Original QQC:**

$$O\left(\sqrt{n}(\log_2(n))^{0.5k}\right)$$

A first inductive algorithm with a quantum query complexity of $\tilde{\Theta}\left(\sqrt{n}\right)$

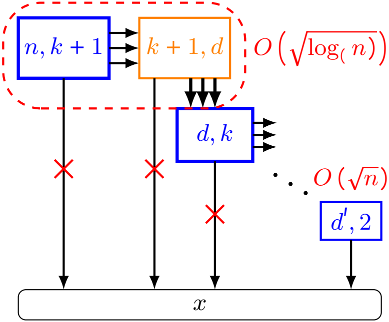


Figure: Schema of the idea of the original algorithm.

- **Original QQC:**

$$O\left(\sqrt{n}(\log_2(n))^{0.5k}\right)$$

- **Small revision:**

$$O\left(\sqrt{n}(\log_2(n))^{0.5(k-1)}\right)$$

small revision

the new algorithm

can be plug in the big one

Sommaire

Conclusion

What as been done:

-

Possible idea to go further:

-