

Computação Quântica

Aula 17

Murilo V. G. da Silva

DINF/UFPR

O Algoritmo de Grover

Objetivo: encontrar um elemento de um conjunto de tamanho N em tempo $\mathcal{O}(\sqrt{N})$.

O Algoritmo de Grover

Objetivo: encontrar um elemento de um conjunto de tamanho N em tempo $\mathcal{O}(\sqrt{N})$.

- Embora seja conhecido com algoritmo de “busca”, não é uma busca no sentido clássico em que estamos acostumados

O Algoritmo de Grover

Objetivo: encontrar um elemento de um conjunto de tamanho N em tempo $\mathcal{O}(\sqrt{N})$.

- Embora seja conhecido com algoritmo de “busca”, não é uma busca no sentido clássico em que estamos acostumados
(aqui não é fornecido um vetor de entrada para encontrar um elemento via busca binária, sequencial, etc.)

O Algoritmo de Grover

Objetivo: encontrar um elemento de um conjunto de tamanho N em tempo $\mathcal{O}(\sqrt{N})$.

- Embora seja conhecido com algoritmo de “busca”, não é uma busca no sentido clássico em que estamos acostumados
(aqui não é fornecido um vetor de entrada para encontrar um elemento via busca binária, sequencial, etc.)
- Aqui busca significa encontrar um objeto que tem certa propriedade

O Algoritmo de Grover

Objetivo: encontrar um elemento de um conjunto de tamanho N em tempo $\mathcal{O}(\sqrt{N})$.

- Embora seja conhecido com algoritmo de “busca”, não é uma busca no sentido clássico em que estamos acostumados
(aqui não é fornecido um vetor de entrada para encontrar um elemento via busca binária, sequencial, etc.)
- Aqui busca significa encontrar um objeto que tem certa propriedade
- Esta propriedade é a entrada do algoritmo

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?
 - Em teoria da computação uma propriedade é definida por um conjunto de strings (tipicamente um conjunto de strings que compartilham alguma característica em comum), ou seja uma linguagem.

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?
 - Em teoria da computação uma propriedade é definida por um conjunto de strings (tipicamente um conjunto de strings que compartilham alguma característica em comum), ou seja uma linguagem.
- Obviamente estamos preocupados com linguagens (propriedades) decidíveis.

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?
 - Em teoria da computação uma propriedade é definida por um conjunto de strings (tipicamente um conjunto de strings que compartilham alguma característica em comum), ou seja uma linguagem.
- Obviamente estamos preocupados com linguagens (propriedades) decidíveis.
 - Não estamos preocupados com propriedades indecidíveis, ou seja, dada um string não existe sequer um algoritmo que consegue decidir se a dada string possui ou não tal propriedade. Em outras palavras, estamos preocupados com linguagens recursivas. Ou seja, a propriedade (linguagem) em questão é totalmente definida pela máquina de Turing que a decide.

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?
 - Em teoria da computação uma propriedade é definida por um conjunto de strings (tipicamente um conjunto de strings que compartilham alguma característica em comum), ou seja uma linguagem.
- Obviamente estamos preocupados com linguagens (propriedades) decidíveis.
 - Não estamos preocupados com propriedades indecidíveis, ou seja, dada um string não existe sequer um algoritmo que consegue decidir se a dada string possui ou não tal propriedade. Em outras palavras, estamos preocupados com linguagens recursivas. Ou seja, a propriedade (linguagem) em questão é totalmente definida pela máquina de Turing que a decide.
- Para o algoritmo de Grover, a entrada é a propriedade (i.e., um algoritmo).

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?
 - Em teoria da computação uma propriedade é definida por um conjunto de strings (tipicamente um conjunto de strings que compartilham alguma característica em comum), ou seja uma linguagem.
- Obviamente estamos preocupados com linguagens (propriedades) decidíveis.
 - Não estamos preocupados com propriedades indecidíveis, ou seja, dada um string não existe sequer um algoritmo que consegue decidir se a dada string possui ou não tal propriedade. Em outras palavras, estamos preocupados com linguagens recursivas. Ou seja, a propriedade (linguagem) em questão é totalmente definida pela máquina de Turing que a decide.
- Para o algoritmo de Grover, a entrada é a propriedade (i.e., um algoritmo).
- Aqui estaremos preocupados apenas com o caso mais difícil, que é o caso onde apenas uma string possui tal propriedade (ou seja, a linguagem tem apenas um elemento).

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?
 - Em teoria da computação uma propriedade é definida por um conjunto de strings (tipicamente um conjunto de strings que compartilham alguma característica em comum), ou seja uma linguagem.
- Obviamente estamos preocupados com linguagens (propriedades) decidíveis.
 - Não estamos preocupados com propriedades indecidíveis, ou seja, dada um string não existe sequer um algoritmo que consegue decidir se a dada string possui ou não tal propriedade. Em outras palavras, estamos preocupados com linguagens recursivas. Ou seja, a propriedade (linguagem) em questão é totalmente definida pela máquina de Turing que a decide.
- Para o algoritmo de Grover, a entrada é a propriedade (i.e., um algoritmo).
- Aqui estaremos preocupados apenas com o caso mais difícil, que é o caso onde apenas uma string possui tal propriedade (ou seja, a linguagem tem apenas um elemento).
- Note que neste caso trata-se de um algoritmo que aceita apenas uma string dentre as 2^n strings possíveis (uma agulha em um palheiro).

O Algoritmo de Grover

Busca por objeto com certa propriedade

Dada uma propriedade, encontrar um objeto de tamanho n (uma string de tamanho n que representa o objeto) sem precisar enumerar as $2^n = N$ strings de tamanho n .

- O que significa “uma dada propriedade”?
 - Em teoria da computação uma propriedade é definida por um conjunto de strings (tipicamente um conjunto de strings que compartilham alguma característica em comum), ou seja uma linguagem.
- Obviamente estamos preocupados com linguagens (propriedades) decidíveis.
 - Não estamos preocupados com propriedades indecidíveis, ou seja, dada um string não existe sequer um algoritmo que consegue decidir se a dada string possui ou não tal propriedade. Em outras palavras, estamos preocupados com linguagens recursivas. Ou seja, a propriedade (linguagem) em questão é totalmente definida pela máquina de Turing que a decide.
- Para o algoritmo de Grover, a entrada é a propriedade (i.e., um algoritmo).
- Aqui estaremos preocupados apenas com o caso mais difícil, que é o caso onde apenas uma string possui tal propriedade (ou seja, a linguagem tem apenas um elemento).
- Note que neste caso trata-se de um algoritmo que aceita apenas uma string dentre as 2^n strings possíveis (uma agulha em um palheiro).

No modelo “caixa preta” (que é a maneira como vamos tratar o algoritmo de entrada) o Algoritmo de Grover é ótimo.

O Algoritmo de Grover

Exemplos de agulhas no palheiro (supondo apenas uma agulha):

O Algoritmo de Grover

Exemplos de agulhas no palheiro (supondo apenas uma agulha):

- Dado uma fórmula ϕ em 3-CNF, encontrar uma valoração que a satisfaça.

O Algoritmo de Grover

Exemplos de agulhas no palheiro (supondo apenas uma agulha):

- Dado uma fórmula ϕ em 3-CNF, encontrar uma valoração que a satisfaça.
- Dado um grafo, encontrar um circuito hamiltoniano.

O Algoritmo de Grover

Exemplos de agulhas no palheiro (supondo apenas uma agulha):

- Dado uma fórmula ϕ em 3-CNF, encontrar uma valoração que a satisfaça.
- Dado um grafo, encontrar um circuito hamiltoniano.
- Dado um grafo com pesos, encontrar um circuito que visite todos os vértices cujo peso do circuito seja mínimo.

O Algoritmo de Grover

Exemplos de agulhas no palheiro (supondo apenas uma agulha):

- Dado uma fórmula ϕ em 3-CNF, encontrar uma valoração que a satisfaça.
- Dado um grafo, encontrar um circuito hamiltoniano.
- Dado um grafo com pesos, encontrar um circuito que visite todos os vértices cujo peso do circuito seja mínimo.
- Dada uma instância do problema da mochila, encontrar uma solução.

O Algoritmo de Grover

Exemplos de agulhas no palheiro (supondo apenas uma agulha):

- Dado uma fórmula ϕ em 3-CNF, encontrar uma valoração que a satisfaça.
- Dado um grafo, encontrar um circuito hamiltoniano.
- Dado um grafo com pesos, encontrar um circuito que visite todos os vértices cujo peso do circuito seja mínimo.
- Dada uma instância do problema da mochila, encontrar uma solução.

Ideia: Colocar todas strings em superposição, jogar no algoritmo que testa a propriedade (consequentemente, saída temos cada string com sua saída em superposição) e, de alguma maneira, manipular a superposição para que a “agulha” (string que procuramos) que estamos buscando no palheiro tenha amplitude muito alta.

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) *Inversão de fase*.

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$
(ou seja, $\forall x \in \{0, 1\}^n \setminus \{x'\}, f(x) = 0$)

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$
(ou seja, $\forall x \in \{0, 1\}^n \setminus \{x'\}, f(x) = 0$)

- (1) $\sum_x \alpha_x |x\rangle \implies \sum_{x \neq x'} \alpha_x |x\rangle - \alpha_{x'} |x'\rangle.$

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$
(ou seja, $\forall x \in \{0, 1\}^n \setminus \{x'\}, f(x) = 0$)

- (1) $\sum_x \alpha_x |x\rangle \implies \sum_{x \neq x'} \alpha_x |x\rangle - \alpha_{x'} |x'\rangle.$
- (2) $\sum_x \alpha_x |x\rangle \implies \sum_x (2\mu - \alpha_x) |x\rangle.$

onde $\mu = \sum_x \alpha_x / N$ é a média das amplitudes.

Observe que $(2\mu - \alpha_x) = \mu + (\mu - \alpha_x)$

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$
(ou seja, $\forall x \in \{0, 1\}^n \setminus \{x'\}, f(x) = 0$)

- (1) $\sum_x \alpha_x |x\rangle \implies \sum_{x \neq x'} \alpha_x |x\rangle - \alpha_{x'} |x'\rangle.$
- (2) $\sum_x \alpha_x |x\rangle \implies \sum_x (2\mu - \alpha_x) |x\rangle.$

onde $\mu = \sum_x \alpha_x / N$ é a média das amplitudes.

Observe que $(2\mu - \alpha_x) = \mu + (\mu - \alpha_x)$

Ideia do Algoritmo

Vamos rodar os passos (1) e (2) alternadamente $O(\sqrt{N})$ vezes e no final medir.

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$
(ou seja, $\forall x \in \{0, 1\}^n \setminus \{x'\}, f(x) = 0$)

- (1) $\sum_x \alpha_x |x\rangle \implies \sum_{x \neq x'} \alpha_x |x\rangle - \alpha_{x'} |x'\rangle.$
- (2) $\sum_x \alpha_x |x\rangle \implies \sum_x (2\mu - \alpha_x) |x\rangle.$

onde $\mu = \sum_x \alpha_x / N$ é a média das amplitudes.

Observe que $(2\mu - \alpha_x) = \mu + (\mu - \alpha_x)$

Ideia do Algoritmo

Vamos rodar os passos (1) e (2) alternadamente $O(\sqrt{N})$ vezes e no final medir.

- Veremos que a probabilidade de obter x' é $1/2$

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$
(ou seja, $\forall x \in \{0, 1\}^n \setminus \{x'\}, f(x) = 0$)

- (1) $\sum_x \alpha_x |x\rangle \implies \sum_{x \neq x'} \alpha_x |x\rangle - \alpha_{x'} |x'\rangle.$
- (2) $\sum_x \alpha_x |x\rangle \implies \sum_x (2\mu - \alpha_x) |x\rangle.$

onde $\mu = \sum_x \alpha_x / N$ é a média das amplitudes.

Observe que $(2\mu - \alpha_x) = \mu + (\mu - \alpha_x)$

Ideia do Algoritmo

Vamos rodar os passos (1) e (2) alternadamente $O(\sqrt{N})$ vezes e no final medir.

- Veremos que a probabilidade de obter x' é $1/2$
- Rodamos o algoritmo várias vezes e escolhemos o resultado mais obtido.

O Algoritmo de Grover

Dois passos básicos usados no algoritmo de Grover:

- (1) Inversão de fase.
- (2) Reflexão em torno da média.

Vamos ver mais especificamente o que os passos acima fazem:

Seja f a função que representa a propriedade de entrada (no modelo caixa preta)

Seja x' a única string tal que $f(x') = 1$
(ou seja, $\forall x \in \{0, 1\}^n \setminus \{x'\}, f(x) = 0$)

- (1) $\sum_x \alpha_x |x\rangle \implies \sum_{x \neq x'} \alpha_x |x\rangle - \alpha_{x'} |x'\rangle.$
- (2) $\sum_x \alpha_x |x\rangle \implies \sum_x (2\mu - \alpha_x) |x\rangle.$

onde $\mu = \sum_x \alpha_x / N$ é a média das amplitudes.

Observe que $(2\mu - \alpha_x) = \mu + (\mu - \alpha_x)$

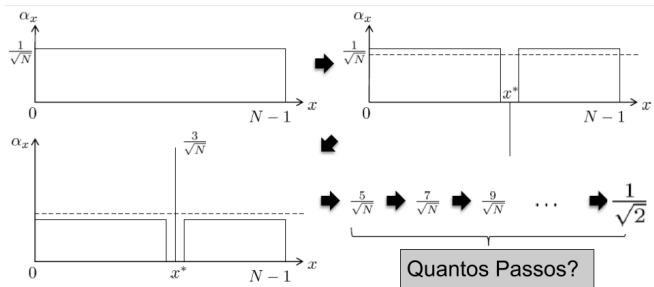
Ideia do Algoritmo

Vamos rodar os passos (1) e (2) alternadamente $O(\sqrt{N})$ vezes e no final medir.

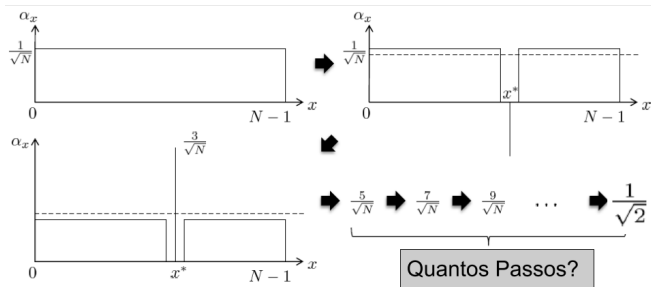
- Veremos que a probabilidade de obter x' é $1/2$
- Rodamos o algoritmo várias vezes e escolhemos o resultado mais obtido.

Observe que se rodarmos o algoritmo 100 vezes, o número esperado de vezes que obtemos x' é 50 e para cada outro x o número esperado de ocorrer (mesmo uma única vez) é exponencialmente pequeno). A probabilidade de que qualquer outro x ocorra 50 vezes é desprezível.

Algoritmo de Grover: Ideia

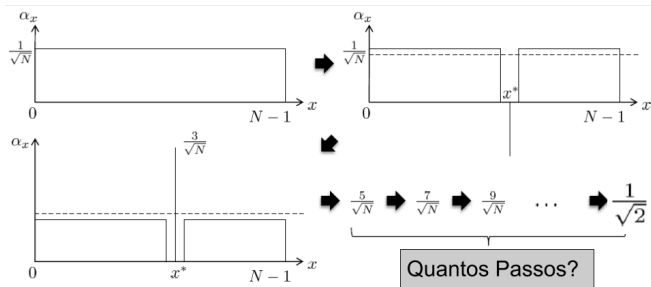


Algoritmo de Grover: Ideia



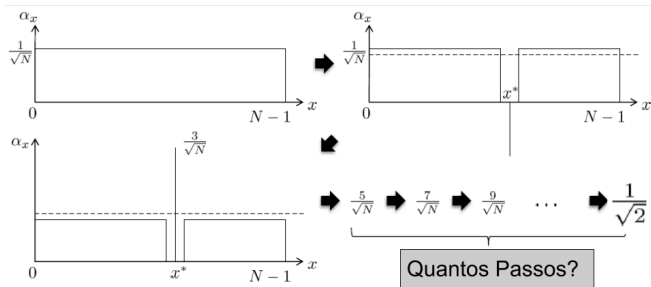
- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.

Algoritmo de Grover: Ideia



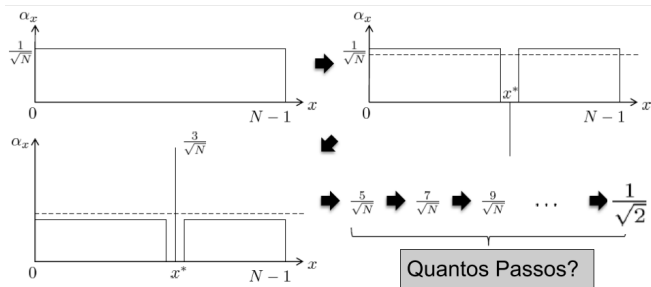
- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.
- O que precisamos analisar:

Algoritmo de Grover: Ideia



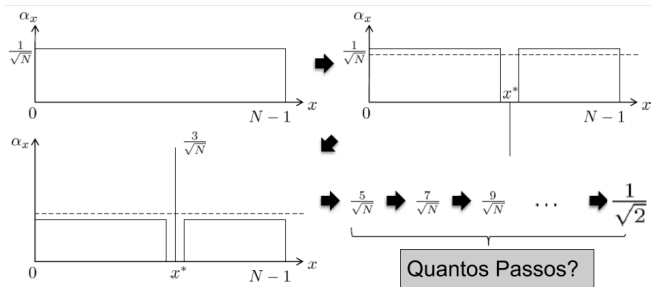
- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.
- O que precisamos analisar:
 - O ganho da amplitude de x^* a cada passo (veremos a seguir)

Algoritmo de Grover: Ideia



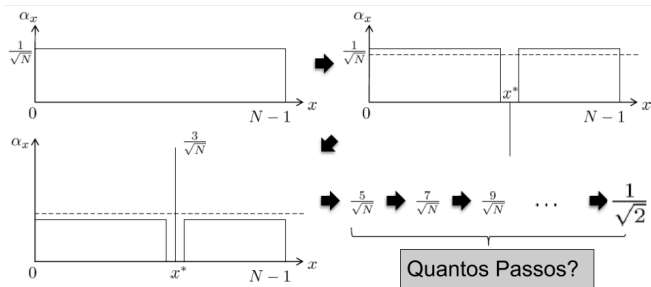
- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.
- O que precisamos analisar:
 - O ganho da amplitude de x^* a cada passo (veremos a seguir)
(a figura sugere $2/\sqrt{N}$, mas na realidade é um pouco menos do que isso)

Algoritmo de Grover: Ideia



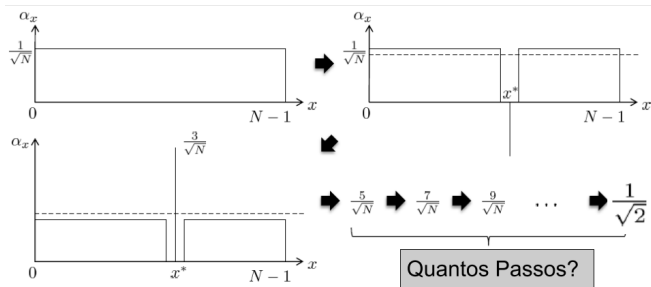
- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.
- O que precisamos analisar:
 - O ganho da amplitude de x^* a cada passo (veremos a seguir)
(a figura sugere $2/\sqrt{N}$, mas na realidade é um pouco menos do que isso)
 - O número de passos (veremos a seguir)

Algoritmo de Grover: Ideia



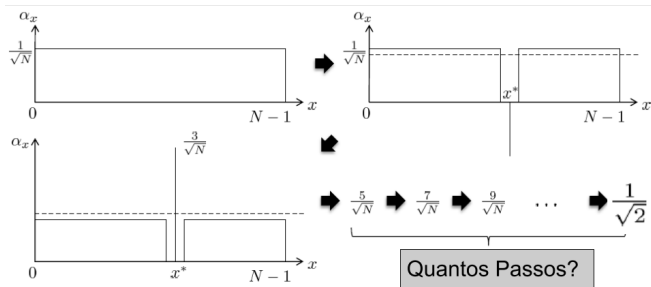
- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.
- O que precisamos analisar:
 - O ganho da amplitude de x^* a cada passo (veremos a seguir)
(a figura sugere $2/\sqrt{N}$, mas na realidade é um pouco menos do que isso)
 - O número de passos (veremos a seguir)
(se a figura estivesse correta, bastariam $\frac{\sqrt{N}}{2\sqrt{2}}$ passos)

Algoritmo de Grover: Ideia



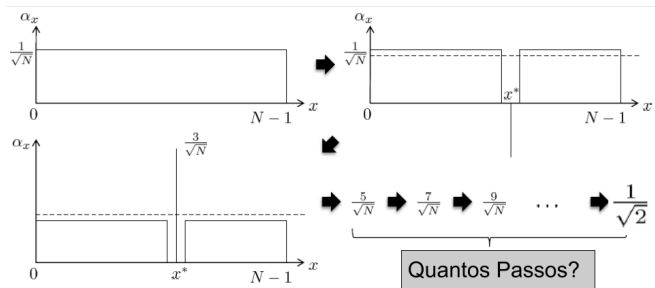
- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.
- O que precisamos analisar:
 - O ganho da amplitude de x^* a cada passo (veremos a seguir)
(a figura sugere $2/\sqrt{N}$, mas na realidade é um pouco menos do que isso)
 - O número de passos (veremos a seguir)
(se a figura estivesse correta, bastariam $\frac{\sqrt{N}}{2\sqrt{2}}$ passos)
 - Como fazer a inversão de fase (Fácil? Alguém sabe como?)

Algoritmo de Grover: Ideia

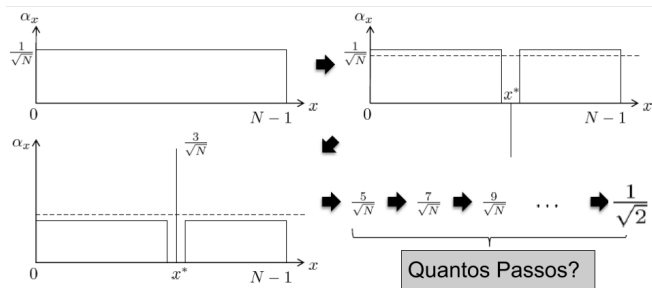


- Note que uma vez que o algoritmo atinge o ponto em que a amplitude de x' é $1/\sqrt{2}$, a probabilidade de obter x' ao medir o sistema é $1/2$.
- O que precisamos analisar:
 - O ganho da amplitude de x^* a cada passo (veremos a seguir)
(a figura sugere $2/\sqrt{N}$, mas na realidade é um pouco menos do que isso)
 - O número de passos (veremos a seguir)
(se a figura estivesse correta, bastariam $\frac{\sqrt{N}}{2\sqrt{2}}$ passos)
 - Como fazer a inversão de fase (Fácil? Alguém sabe como?)
 - Como fazer a reflexão sobre a média (Aula que vem)

Algoritmo de Grover: Incremento de x^* e número de passos

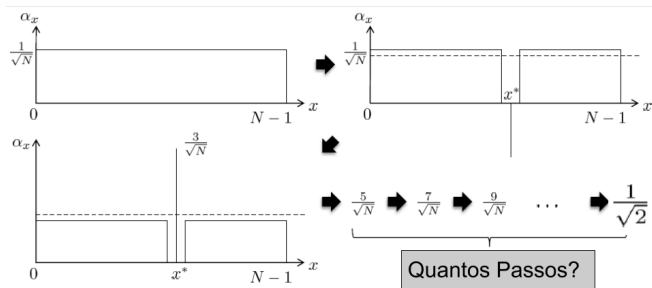


Algoritmo de Grover: Incremento de x^* e número de passos



- Fazendo uma análise mais fina, a cada passo a amplitude de x' aumenta no mínimo $\sqrt{2/N}$ a cada passo. Por quê?

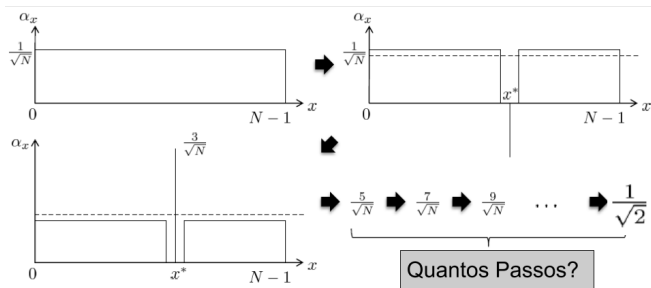
Algoritmo de Grover: Incremento de x^* e número de passos



- Fazendo uma análise mais fina, a cada passo a amplitude de x' aumenta no mínimo $\sqrt{2/N}$ a cada passo. Por quê?

No pior caso α'_x já atingiu $\frac{1}{\sqrt{2}}$ e o restante das amplitudes deve estar distribuído igualmente.

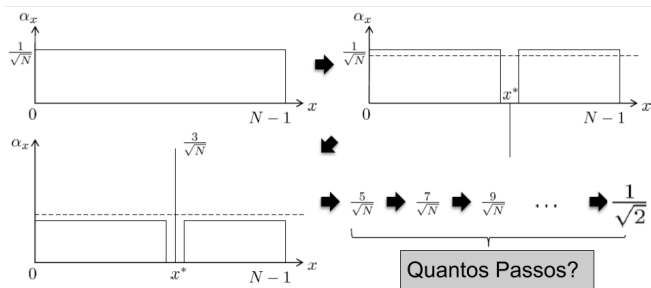
Algoritmo de Grover: Incremento de x^* e número de passos



- Fazendo uma análise mais fina, a cada passo a amplitude de x' aumenta no mínimo $\sqrt{2/N}$ a cada passo. Por quê?

No pior caso $\alpha_{x'}$ já atingiu $\frac{1}{\sqrt{2}}$ e o restante das amplitudes deve estar distribuído igualmente. Portanto cada demais x vale $1/\sqrt{2(N-1)}$. Para simplificar e sem perder a validade do argumento, digamos que que cada $\alpha_x \geq 1/\sqrt{2N}$.

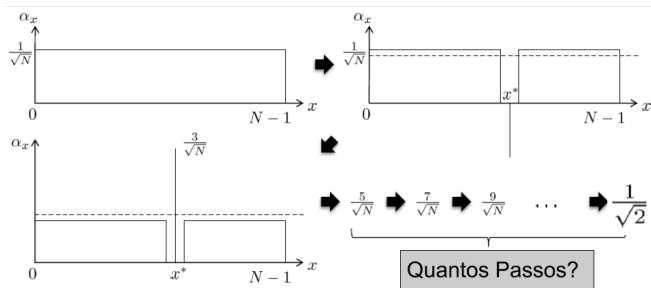
Algoritmo de Grover: Incremento de x^* e número de passos



- Fazendo uma análise mais fina, a cada passo a amplitude de x' aumenta no mínimo $\sqrt{2/N}$ a cada passo. Por quê?

No pior caso α'_x já atingiu $\frac{1}{\sqrt{2}}$ e o restante das amplitudes deve estar distribuído igualmente. Portanto cada demais x vale $1/\sqrt{2(N-1)}$. Para simplificar e sem perder a validade do argumento, digamos que que cada $\alpha_x \geq 1/\sqrt{2N}$. Com isso a variação de $\alpha'_x \geq 2/\sqrt{2N} = \sqrt{2/N}$ a cada passo.)

Algoritmo de Grover: Incremento de x^* e número de passos

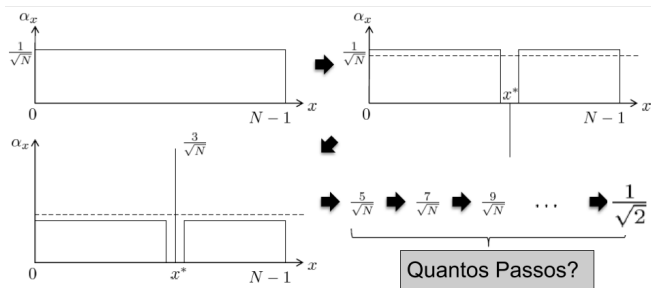


- Fazendo uma análise mais fina, a cada passo a amplitude de x' aumenta no mínimo $\sqrt{2/N}$ a cada passo. Por quê?

No pior caso α'_x já atingiu $\frac{1}{\sqrt{2}}$ e o restante das amplitudes deve estar distribuído igualmente. Portanto cada demais x vale $1/\sqrt{2(N-1)}$. Para simplificar e sem perder a validade do argumento, digamos que que cada $\alpha_x \geq 1/\sqrt{2N}$. Com isso a variação de $\alpha'_x \geq 2/\sqrt{2N} = \sqrt{2/N}$ a cada passo.)

- Com isso garantimos que $\mathcal{O}(\sqrt{N})$ passos é o suficiente para α'_x atinja $1/\sqrt{2}$

Algoritmo de Grover: Incremento de x^* e número de passos



- Fazendo uma análise mais fina, a cada passo a amplitude de x' aumenta no mínimo $\sqrt{2/N}$ a cada passo. Por quê?

No pior caso α'_x já atingiu $\frac{1}{\sqrt{2}}$ e o restante das amplitudes deve estar distribuído igualmente. Portanto cada demais x vale $1/\sqrt{2(N-1)}$. Para simplificar e sem perder a validade do argumento, digamos que que cada $\alpha_x \geq 1/\sqrt{2N}$. Com isso a variação de $\alpha'_x \geq 2/\sqrt{2N} = \sqrt{2/N}$ a cada passo.)

- Com isso garantimos que $\mathcal{O}(\sqrt{N})$ passos é o suficiente para α'_x atinja $1/\sqrt{2}$ (pois $\frac{1}{2}\sqrt{N}$ passos vezes o incremento $\sqrt{2/N}$ é igual a $1/\sqrt{2}$.)