Lecture 18 – Grover's Algorithm

[Last lecture was a bit abstract. Today we'll get more concrete & see the 2nd-most famous quantum algorithm – Grover's Alg. for unstructured search.]

Task: Given N bits, find a 1.

usually implicitly: truth table of Boolean function/circuit $C:\{0,1\}^n \to \{0,1\}$. $(N=2^n)$

[[The old "finding 'patterns' in implicitly rep'd data" task. Except this time – there aren't really "patterns". Just "finding".]]

In the "black-box query" model...

[[where you're not allowed to 'analyze' C, just use its I/O behavior]]

- Deterministic alg: needs N queries
- Randomized alg: $\gtrsim N$ queries. E.g., $\geq \frac{N}{2}$ on average to have 50% chance of success. [[Imagine there's only one x with $F(x)=1$.]]
- Quantum alg: ... $\lessapprox \sqrt{N}$ queries suffice [Grover '96]

Given description of circuit C, this is precisely the famous (CIRCUIT-) SAT problem.

"NP-complete". [[like, you can try to 'analyze' C.]]

"$P \neq NP$" $\Leftrightarrow$ no poly(n)-time classical alg.

"SETH" (Strong Expon. Time Hypoth.) $\Rightarrow$ no 1.9999^n-time alg.

[[basically $\Rightarrow 2^n$-time, i.e., brute-force, is required]]

[[Would be true if you believed circuits could be "obfuscated".]] $\rightarrow$ [[fairly well-believed; also, believed that randomized algs. don't really help]]

Today: $\approx \sqrt{N} = \sqrt{2^n} = \sqrt{2}^n \leq 1.42^n$ time suffices on a quantum computer

[[Actually, we'll focus on fact that $\approx \sqrt{N}$ applications of Q_C suffice w/ high prob., but the alg. is o/w efficient, too. (Unlike the [ETH] HSP solution from last time.

Formally, $O(\#\text{ gates in } C + n) \cdot \sqrt{2}^n$ quantum gates suffice.]]

$\rightarrow$ Still expon. time for SAT, but defies SETH!

[[Could one do better? It would be super-extraordinary to solve SAT on a quantum computer in poly(n), $2^{\sqrt{n}}$, or even 1.01^n time...]]

(Yes, they proved this lower bound before Grover proved his upper bound!!)

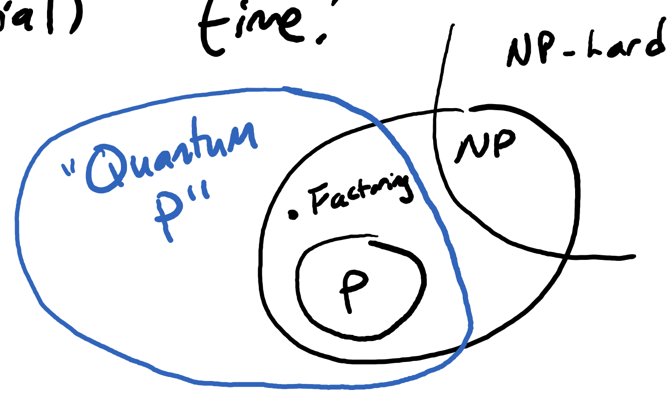
thm: [BBBV'94] In the "black-box query" model, $\gtrsim \sqrt{N}$ applications of Q_c are <u>needed</u>.

[[So Grover's analysis in this model is <u>tight</u>.]]

[[This <u>doesn't</u> prove that Q.C.'s can't solve SAT faster than $\sqrt{2}^n$, just as the fact that "classical black-box query algs need $\gtrsim N$ queries" doesn't prove SETH. Still, it's...]]

(*) <u>evidence</u> that Q.C.'s cannot solve NP-complete problems in polynomial (even subexponential) time!

[[believed worldview →]]



[[Time to starting explaining Grover's Alg.!]]

Assume you're given quantum circuit Q_F 'implementing' some Boolean func. $F:\{0,1\}^n \to \{0,1\}$

Want to find $x \in \{0,1\}^n$ s.t. $F(x)=1$.

[[Or else become confident none exists, if $F \equiv 0$.]]

[[Recall, by the way, that given a classical circuit C computing F – as in the SAT problem – it's easy & efficient to convert it to Q_F.]]

Key difference from B.V. / Simon / Shor....

(*) F <u>not</u> assumed to have any special structure/pattern

[[Can literally be anything. Versus, say, Bernstein–Vazirani, where we're promised $F(x) = XOR_s(x)$ for some s, or Simon, where $F(x) = F(x+s)$ $\forall x$.

It is generally believed that this "lack of structure" is why Grover only gets polynomial, not exponential, speedup.]]

[Having said all that, I'll now impose a promise on F, for simplicity! ☺]

Assume (for now): $F(x) = 1$ for exactly one string, call it $x^* \in \{0,1\}^n$.

Task: Find x^*.

This is the hardest case.

[I assure you. The more 1's F has in its truth table, the easier it should be to find them. We'll see this is in fact true; if F has K 1's in its truth table, can find one in $\leq \sqrt{N/K}$ queries! All variants of Grover's Alg reduce to this "unique 1" case.]

[Since F outputs only one bit, we'll also use the...]

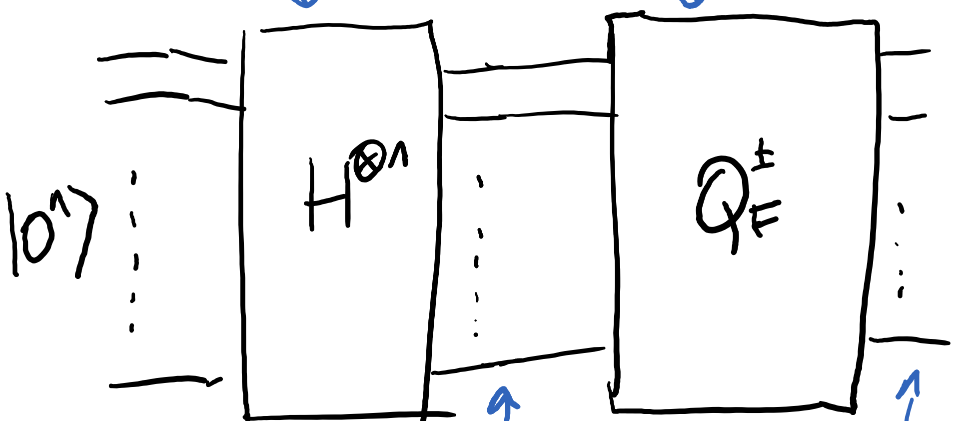
Sign-implem. trick: use $Q_F^{\pm}$, maps $|x\rangle \longmapsto (-1)^{F(x)}|x\rangle$.

$$\text{ie.} \quad \begin{bmatrix} \alpha_{00\cdots0} \\ \alpha_{00\cdots1} \\ \vdots \\ \alpha_{x^*} \\ \vdots \\ \alpha_{11\cdots1} \end{bmatrix} \xmapsto{Q_F^{\pm}} \begin{bmatrix} \alpha_{00\cdots0} \\ \alpha_{00\cdots1} \\ \vdots \\ -\alpha_{x^*} \\ \vdots \\ \alpha_{11\cdots1} \end{bmatrix}$$

[Just negates one "secret" amplitude.]

[So. What to do?]

Rotate - Compute - Rotate paradigm?



??

unif. superpos.

$\frac{1}{\sqrt{N}} \sum_x |x\rangle$

$\frac{1}{\sqrt{N}} \sum_x (-1)^{F(x)} |x\rangle$

$$= \frac{1}{\sqrt{N}} \begin{bmatrix} +1 \\ \vdots \\ +1 \\ -1 \\ +1 \\ \vdots \\ +1 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow x^{*}\text{'th} \\ \\ \\ \\ \end{matrix}$$

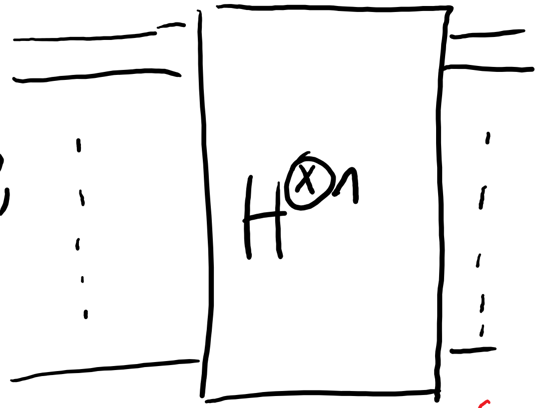
Measuring now clearly bad.
Get each "x" with prob. $\frac{1}{N}$.
<u>No</u> info about x^*!

[Since no inherent structure to domain, may as well try the simplest...]

Hadamard
Fourier
Transformation?

"$|f\rangle$" $= \frac{1}{\sqrt{N}} \sum_x f(x)|x\rangle$,

$f(x) = (-1)^{F(x)}$



$$\sum_{s \in \{0,1\}^n} \hat{f}(s)\,|s\rangle$$

["strength of XORs pattern"]

$$\underset{x \in \{0,1\}^n}{\text{avg}} \{ f(x) \cdot XOR_s(x) \}$$

(*) E.g.: amplitude on $|0^n\rangle$ $(s = 00\cdots0)$

is $(\because XOR_s(x) = (-1)^{0 \cdot x} = 1)$ ...

$$\underset{x \in \{0,1\}^n}{\text{avg}} \{ f(x) \} =: \text{"}\mu\text{"} \quad \text{"mean" of } f.$$

$$= \underbrace{1 - 2/2^n} \overset{!}{\approx} 1.$$

$\begin{bmatrix} \approx 1 \\ \approx 0 \\ \approx 0 \\ \vdots \\ \approx 0 \end{bmatrix} \approx |0^n\rangle$

No surprise: $Q_F^{\pm}$ barely does anything, so circuit is similar to $|0^n\rangle - H^{\otimes n} - H^{\otimes n} -$.

$\Rightarrow$ all other amplitudes are expon. small.

(Fact: $\hat{f}(s) = \pm\, 2/2^n \quad \forall s \neq 0^n$.)

$\Rightarrow$ Measuring now also bad :(. [Except with prob. $\approx 4/2^n$, will just see "$000\cdots0$".]

Grover Idea:

... "Compute/Rotate" again! (and again, and again....)

manipulate the Fourier transform

(Inverse) Fourier transform it back, get a "slight modification of $|f\rangle$" in which $|x^*\rangle$ amplitude slightly more highlighted. Repeat.

$$|f\rangle \xmapsto{H^{\otimes n}} \sum_s \hat{f}(s)|s\rangle, \text{ where}$$

$\hat{f}(s) = \langle \chi_s | f \rangle$ = coeff. of $|f\rangle$ in orthonormal basis of $\{|\chi_s\rangle\}$, $\chi_s(x) = (-1)^{XOR_s(x)}$.

(Recall: $\chi_{0^n} \equiv 1$, $\hat{f}(0^n) = \langle \chi_{0^n} | f \rangle = \frac{1}{N} [1\ 1\ \cdots\ 1] \begin{bmatrix} f(0\cdots0) \\ f(0\cdots1) \\ \vdots \\ f(1\cdots1) \end{bmatrix}$

$= \underset{x}{\text{avg}} \{f(x)\} =:$ "μ".)

And so $|f\rangle = \sum_s \hat{f}(s)\,|\chi_s\rangle$

$$= \mu \cdot |\text{const. 1 func}\rangle + f^{dev}$$

$$f^{dev} := \sum_{s \neq 0^n} \hat{f}(s)\,|\chi_s\rangle.$$

Function viewpoint: $f = \underset{\text{const.}}{\mu} + f^{dev}$

function defined by $|f^{dev}\rangle = \sum_{s \neq 0} \hat{f}(s) |\chi_s\rangle$

Vector viewpoint:

$$\frac{1}{\sqrt{N}}\begin{bmatrix} f(0\ldots0) \\ \vdots \\ f(1\ldots1) \end{bmatrix} = \frac{1}{\sqrt{N}}\begin{bmatrix} \mu \\ \mu \\ \mu \\ \vdots \\ \mu \end{bmatrix} + \frac{1}{\sqrt{N}}\begin{bmatrix} f^{dev}(0\ldots0) \\ \vdots \\ f^{dev}(1\ldots1) \end{bmatrix}$$

entries sum to 0.

Idea: Let $\boxed{f^{new} = \mu - f^{dev}.}$

(*) f^{new} has $\widehat{f^{new}}(s) = \begin{cases} \mu = \hat{f}(0^n) & \text{if } s = 0^n \\ -\hat{f}(s) & \text{if } s \neq 0^n \end{cases}$

Can get to $|f^{new}\rangle$ by:

(where we are now, after R.C.R.)

(0) starting from $H^{\otimes n}|f\rangle = \sum_s \hat{f}(s)|s\rangle$,

"new Compute" → (1) negating all amplitudes except on $|0^n\rangle$

"new Rotate" → (2) "Fourier-transforming back" ($H^{\otimes n}$ again)

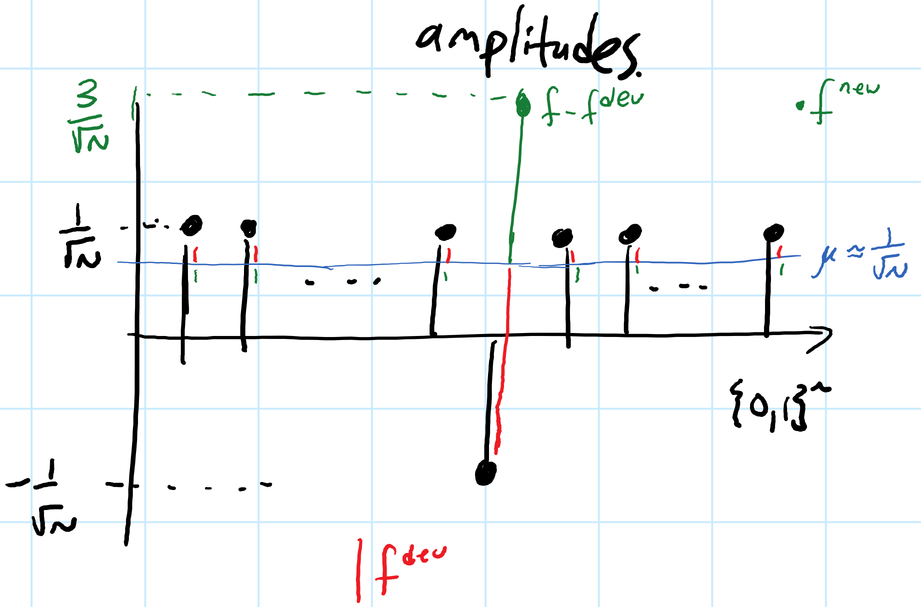
Q1: How to negate all ampls., other than $|0^n\rangle$'s?

Q2: Why is it useful?

Re Q2: "currently"

$$\frac{1}{\sqrt{N}}\begin{bmatrix} +1 \\ \vdots \\ +1 \\ -1 \\ +1 \\ \vdots \\ +1 \end{bmatrix}$$

x^*'th entry



[[We'll analyze it shortly, but this "reflect across mean" operation is very useful.]]

Re Q1: Is this a unitary op.?? Yes,

Clearly preserves (squared) magnitude of states.

Indeed, we're doing $|0^n\rangle \longmapsto |0^n\rangle$

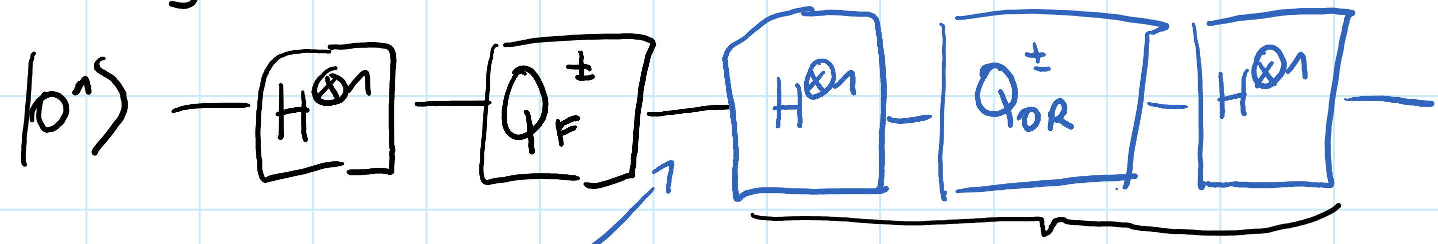
$|s\rangle \longmapsto -|s\rangle$ else.

I.e., $|s\rangle \mapsto (-1)^{OR(s)}|s\rangle$, for logical $OR: \{0,1\}^n \to \{0,1\}$

It's just sign-implementation of OR! $Q^{\pm}_{OR}$.

Certainly $\exists$ efficient ($n-1$ gates) classical circuit for OR $\to$ hence eff. quantum circuit ($O(n)$ gates) for $Q^{\pm}_{OR}$!

Summary:



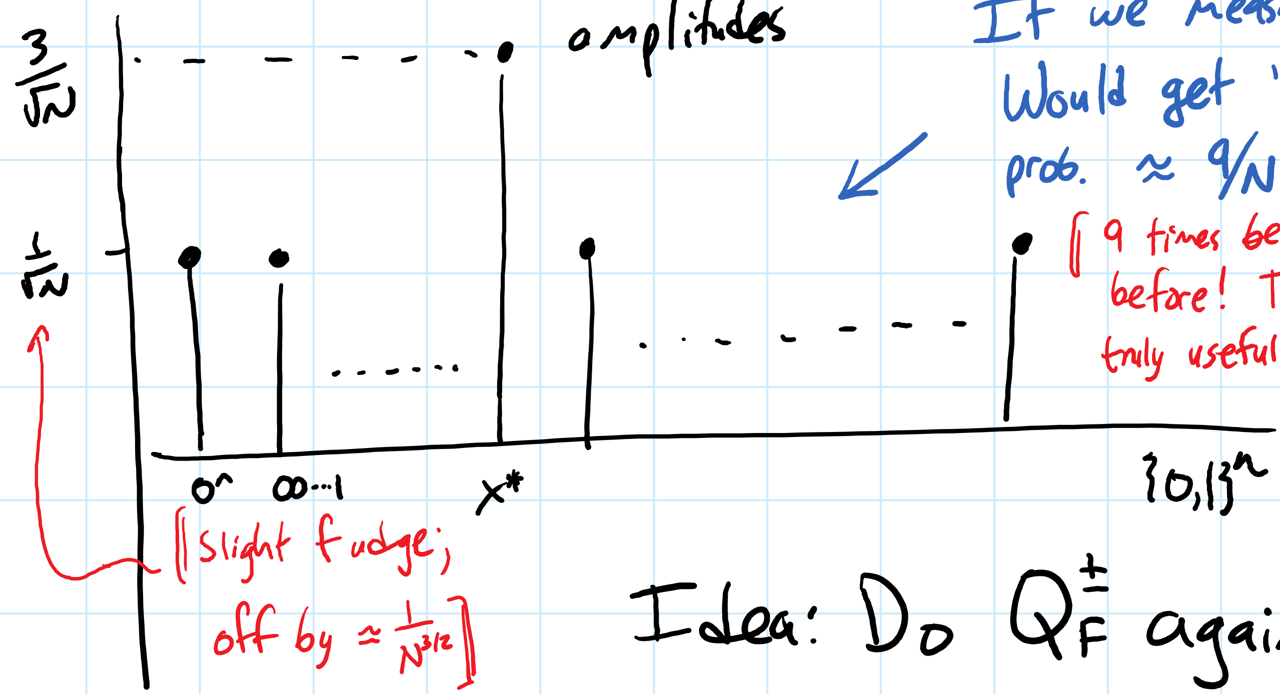
$$\frac{1}{\sqrt{N}}\begin{bmatrix}+1\\ \vdots \\ +1 \\ -1 \\ +1 \\ \vdots \\ +1\end{bmatrix}$$

Grover's "reflect across the mean" operator.

const. $\cdot n$ gates

Mean μ is $\overset{\text{FUDGE}}{\approx} \frac{1}{\sqrt{N}}$ (In truth, $\frac{1}{\sqrt{N}} - \frac{2}{N^{3/2}}$)

After "reflect across mean" op.:

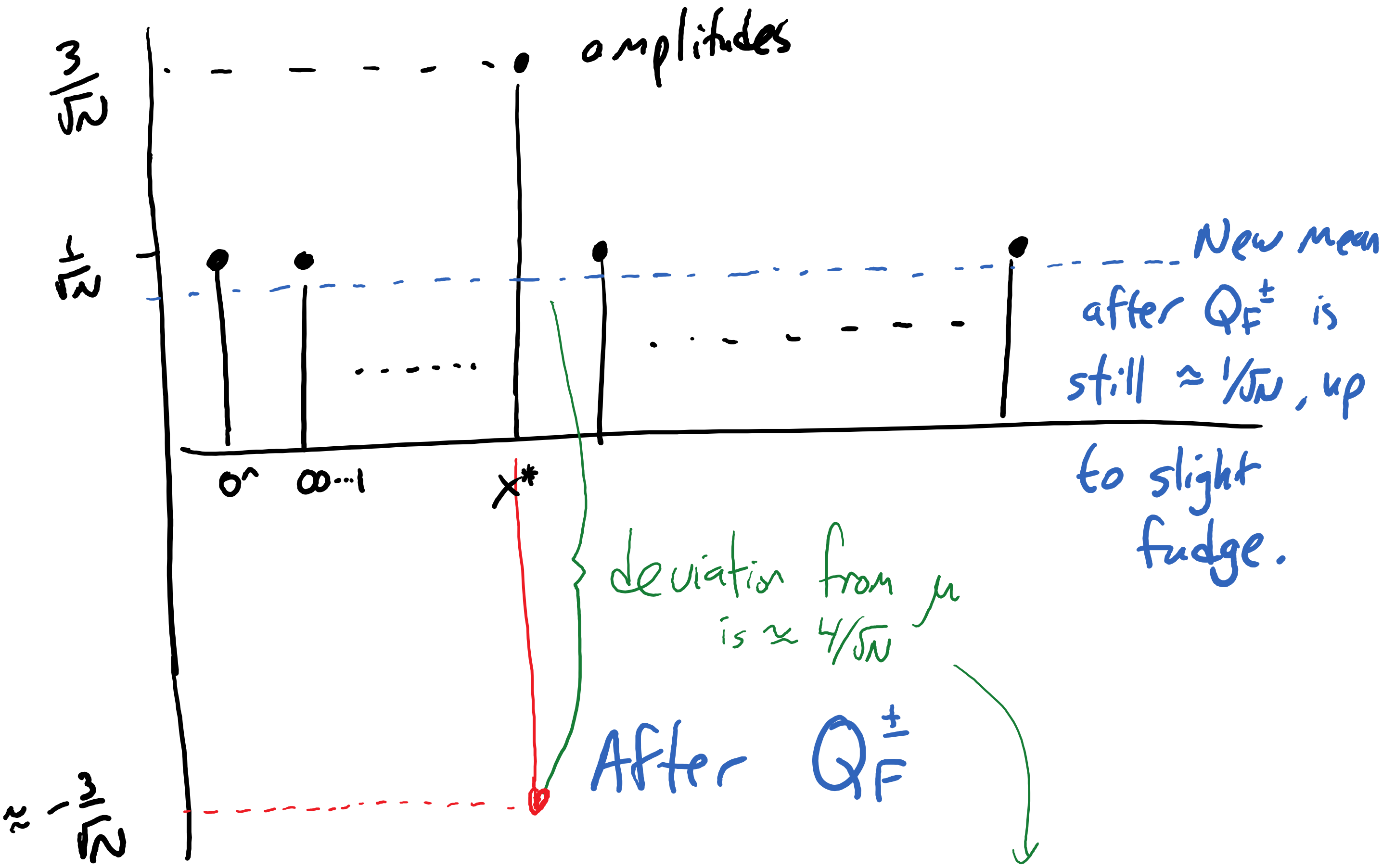


If we measured now: Would get "x^*" with prob. $\approx 9/N$.

[9 times better than before! Tho not truly useful... yet!]

[Slight fudge; off by $\approx \frac{1}{N^{3/2}}$]

Idea: Do $Q_F^{\pm}$ again!



Another "reflect across mean" operation puts $|x^*\rangle$'s amplitude to $\approx 5/\sqrt{N}$.

[[Measure now: see x^* with prob. $\approx 25/N$!]]

$\frac{5}{\sqrt{N}} \xmapsto{Q_F^\pm} -\frac{5}{\sqrt{N}}$, mean $\approx \frac{1}{\sqrt{N}}$ still, dev. from mean $\frac{6}{\sqrt{N}}$

$\xmapsto{H^{\otimes n},\ Q_{OR}^\pm,\ H^{\otimes n}}$ $|x^*\rangle$ has ampl. $\approx \frac{7}{\sqrt{N}}$.

. . . .

<u>Summary</u>: Ignoring slight fudging, each

$Q_F^{\pm}$, $H^{\otimes n} \cdot Q_{OR}^{\pm} \cdot H^{\otimes n}$ iteration

uses size(Q_F) + const $\cdot n$ gates, and

amplitude on $|x^*\rangle$ goes like

$\frac{1}{\sqrt{N}} \to \frac{3}{\sqrt{N}} \to \frac{5}{\sqrt{N}} \to \cdots \to \frac{2T+1}{\sqrt{N}}$ after

T iters.

<u>"Basically"</u>: After $\approx \sqrt{N}/2$ iters, $|x^*\rangle$ amplitude is very high.

Measuring yields "x^*" with quite high probability!!

[Somehow similar to "Elitzur-Vaidman Bomb". $\sqrt{\ }$ somehow saves you.]

<u>Fudging?</u> Easy/boring exercise: amplitude on $|x^*\rangle$ increases by between $\frac{1}{\sqrt{N}}$ & $\frac{2}{\sqrt{N}}$ provided it's $\leq .7$ (& $N \geq 20$).

So do $.35\sqrt{N}$ iters, it'll end between .35 & .7.

$\Rightarrow$ measuring yields "x^*" w. prob $\geq .35^2 > 10\%$.

Repeat if you fail. $\Rightarrow \leq 3.5\sqrt{N}$ total iterations. ∎

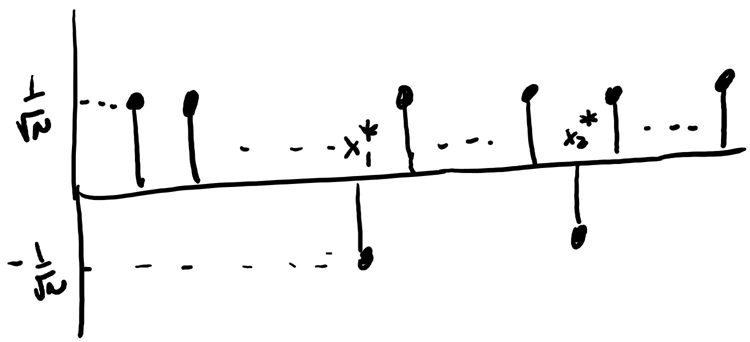
[[Can actually start to go down if you do too many iters.]]

[[If you, like Boyer-Brassard-Høyer-Tapp do it carefully, you find that success prob. becomes ≈ 1 after $\frac{\pi}{4}\sqrt{N}$ iters]]

Extensions - sketched

That was if you were promised F has exactly one 1.

Also works if F has exactly two 1's...



Analysis pretty much the same.

Both ampls ↑ between $\frac{1}{\sqrt{N}}$ & $\frac{2}{\sqrt{N}}$ each iter, provided both $\leq .5$.

After $.25\sqrt{N}$ iters. → ≥ 5% chance of finding x_1^* or x_2^*.

[[Starts to break down, but....]]

Use diff. strategies if you know F has K 1's,

$1 \leq K \leq 2, \quad 2 < K \leq 4, \quad 4 < K \leq 8, \quad \ldots, \quad \frac{N}{4} < K \leq \frac{N}{2}$.

$K > \frac{N}{2}$ → just be classical!

If you know K to w/i factor of 2....

Up to fudging... each of the K "special" amplitudes goes as $\frac{1}{\sqrt{N}} \to \frac{3}{\sqrt{N}} \to \frac{5}{\sqrt{N}} \to \dots$

After T iters, each has ampl. $\approx \frac{2T}{\sqrt{N}}$.

$\therefore$ special ones have collective squared magnitude $\approx 2K\frac{T^2}{N}$.

Fudging OK up till this is $\geq .2$, say

$\to K\frac{T^2}{N} \geq .1 \to T \geq .3\sqrt{\frac{N}{K}}$.

So after $.3\sqrt{\frac{N}{K}}$ iters $\to$ measure, get an x with $F(x)=1$ with $\geq$ constant chance.

Don't know K?

Try $K \approx 1$, then $K \approx 2$, $K \approx 1$

then $K \approx 4$, $K \approx 2$, $K \approx 1$

then $K \approx 8$, $K \approx 4$, $K \approx 2$, $K \approx 1$,

Etc.:

[[Exercise]]

Theorem: Even if $K := \#\{x : F(x) = 1\}$ is unknown...
with $\approx \sqrt{\frac{N}{K}}$ expected queries to $Q_F^{\pm}$ (& $\approx n$ addit'l gates per query) can find some x s.t. $F(x) = 1$...

unless $K = 0$, in which case this will be recognized with high probability after $\approx \sqrt{N}$ queries.