Lecture 19 - Quantum Query Complexity

All quantum algs. we've seen so far [[and, indeed, all (??) cool quantum algs?]] work in the "black-box query model":

"Given" data $F : \{0,1\}^n \to \text{COLORS}$

or $\{0,1,\ldots,N-1\} \to \text{COLORS}$

F	R	G	B	Y	R	G	B	P	R	...
	0	1	2	3	4	......				

via a quantum circuit Q_F for F,

solve some problem "φ" about F.

e.g.: • Simon's problem: $F(x) = F(x+s)$ for some $s \neq 00\cdots0$, all other pairs distinct; find s

• Hidden Subgroup Problem in $(G, \circ)$: $F(x) = F(x \circ h)$ $\forall h \in$ subgroup H, o/w distinct; find H

• Grover's Problem (OR): $\text{COLORS} = \{0,1\}$, find x with $F(x) = 1$ [[or determine no such x]]

[[Observation: none of our algorithms ever "dug into" how Q_F worked. They only applied ("queried") it, in a "black-box fashion".

Further except for the unmentioned nonabelian HSP stuff, all our quantum algs' gate complexity was not much more than "# of times Q_F applied".]]

Motivates...

Query Complexity Model

- Q_F is a "black box", or "oracle"; can <u>only</u> apply it [[don't even imagine it has an "inside" – just a "magic box"]]
- "cost" of an alg. to solve problem "φ": <u>just</u> the # of "queries" = applications of Q_F
- all other computation is "free".

[[So... you can "think about" what x — or superposition of $|x\rangle$'s — to ask Q_F about "as much as you want". Only charged for your questions.]]

. Model variants: Deterministic classical, randomized classical, quantum. [[& "nondeterministic", & others....]]

Why study this model? (Versus the standard gates/time model.)

① All our quantum algs do fit in this model.

② Usually the ignored computational (= non-query) cost is cheap – like, $n^{\text{const}} = (\log N)^{\text{const}}$ gates per query. [[Only exception: that nonabelian HSP stuff...]]

③ Simple enough model, you can prove lower bounds ("no-go / impossibility" theorems)

↳ e.g. next class: Grover problem (OR) requires $\gtrsim \sqrt{N}$ queries to Q_F

④ Gives "evidence" about power of randomization, quantum, nondeterminism, other models ....

New notation for Query Complexity

"Input": ~~function~~ ~~$F : \{0, 1, \ldots, N-1\} \to \text{COLORS}$~~

~~F~~ W:

R	G	B	Y	. . . .
~~$x=0$~~ $i=1$	~~1~~ 2	~~2~~ 3	~~3~~ 4	. . . .

W ↓ string in COLORS^N

$W = RGBY\cdots$

$w_1 = R,\ w_2 = G,\ w_3 = B$ etc.

Now: · classically, you "query" $i \in \{1, 2, \ldots, N\}$, get back w_i

· quantumly, you can query a superposition $\sum_{i=1}^{N} \alpha_i |i\rangle$.

$$Q_F : |i\rangle \otimes |b\rangle \longmapsto |i\rangle \otimes |b \oplus w_i\rangle$$

[[as usual, encoded by some m-bit string]]

We'll also [[for simplicity]] focus on <u>decision</u> (= yes/no) tasks "φ".

E.g.: φ = "Decision - Grover": $\text{COLORS} = \{0, 1\}$ [[usually called "symbols", or "alphabet", actually]]

Input is (unknown) $w \in \{0,1\}^N$.

Can query (superpositions) of $i \in \{1, 2, \ldots, N\}$.

Decide: $\exists ?\ i$ s.t. $w_i = 1$, yes or no.

(That is: compute $\text{OR}(w_1, \ldots, w_N)$.)

e.g.: φ = "Decision-Simon":

Unknown w, eg $w = RGBYRGBYR\cdots$

Can query i, get w_i. [[or superpositions]]

Promise: $\exists$ $s \in \{0,1\}^n$ s.t.

$$w_i = w_j \text{ iff } \text{base2}(i) \oplus \text{base2}(j) = s.$$

Decide: Is $s = 00\cdots0$ ($\Rightarrow$ all entries of w distinct)

or $s \neq 00\cdots0$ ($\Rightarrow$ $N/2$ colors appear in pairs, paired up in special way)

[[Remark: Decision-Simon is no more challenging than the usual "Search-Simon", where you're promised $s \neq 00\cdots0$, have to find s.

Why? Given ability to solve Search-Simon, can solve Decision-Simon like this:

① Run Search-Simon alg. to try to find s.

② Test your candidate s. If it works — say $w_1 = w_j$ where $\text{base2}(j) = 00\cdots01 \oplus s$ — output "$s \neq 00\cdots0$".

③ Else output "$s = 00\cdots0$".]]

Important note: Decision problems φ may be...

"total": every possible input w [in COLORS^N] is allowed

e.g.: Decision-Grover $= \text{OR}(w)$

or "partial/promise": promised that $w \in \mathcal{D} \subseteq \text{COLORS}^N$

a special subset of inputs

e.g. Decision-Simon. $\mathcal{D} =$ all w that are "s-periodic" for some s.

Notation: Let φ be a decision problem (total or partial) about strings w.

$D(\varphi)$: least # of queries needed by a Deterministic alg. (for worst-case input)

$R(\varphi)$: least # of ~ Randomized alg. that, for all w, gets right answer w/ prob. $\geq \frac{2}{3}$.

$Q(\varphi)$: ~ Quantum alg. [can query superpositions] for all w, gets right answer w/ prob $\geq 2/3$

Examples: $\varphi = \text{OR}$ (total) [[Grover's prob.]]

$D(\varphi) = \underline{N}$

$R(\varphi) = \lceil \tfrac{2}{3} N \rceil$ → [[Kind of clear that best alg. is "query w_i for $\tfrac{2}{3}N$ random i – if you see a 1, return "TRUE", else if all 0's, guess "FALSE". For $w = 000\cdots0$, correctness prob. is 100%. If w has at least one 1, correctness prob. $\geq \tfrac{2}{3}$.]]

$Q(\varphi) \lesssim \sqrt{N}$

↓

Grover's Alg.
[[last lecture]]

Next lecture: $Q(\varphi) \gtrsim \sqrt{N}$ [BBBV ca. '96]

$\varphi =$ Decision-Simon: (Partial)

- $D(\varphi)$, $R(\varphi)$ are proportional to $\sqrt{N}$

  [[For $D(\varphi) \lesssim \sqrt{N}$: query all w_i where base2(i) starts w/ $\frac{n}{2}$ 0's and all " " ends " " ". If $s \neq 0$, you'll hit a match between $s_1 s_2 \ldots s_{n/2} 00\cdots0$, $00\cdots0 s_{\frac{n}{2}+1} \cdots s_n$]]

- $Q(\varphi) \lesssim$ "n" $= \log N$.

[[Rem: if you set it up so that "YES" = all distinct, "NO" = $s \neq 0$, then even <u>nondeterministic</u> query complexity is $\gtrsim \sqrt{N}$, I think.]]

[[Exponential gap! Can show $\gtrsim \log N$, too.]]

Examples: φ = "2-to-1 problem" (aka "Collision")
(Partial.)

Input $w \in \text{COLORS}^N$, e.g. $w = RGPBRPBG$

Promise: (i) either all colors distinct; or

(ii) each color used is used exactly 2 times

(but there's no pattern to the pairing)

[Have to decide which.]

$D(\varphi) = \frac{N}{2}+1$ $\qquad R(\varphi) \approx \sqrt{N}$ [Birthday Attack].

claim: $Q(\varphi) \lesssim N^{1/3}$. $\qquad$ (Harder thm: [AS'02]: $Q(\varphi) \gtrsim N^{1/3}$.)

sketch: ① Pick subset L of $N^{1/3}$ coordinates, at random

② Query w_i $\forall i \in L$ (cost: $N^{1/3}$). Call answers A.

If A has duplicates $\rightarrow$ in case (ii). Else...

③ Make new oracle Q' that, given $j \notin L$,

returns 1 if $w_j \in A$, 0 else.

Q' only needs to use Q once.

④ Do Grover on Q'!

Case (i): Q' always returns 0.

Case (ii): Q' returns 1 on $K = N^{1/3}$ j's.

Grover cost: $\sqrt{\frac{N - N^{1/3}}{K}} \approx \sqrt{N^{2/3}} = N^{1/3}$. ∎

Examples: φ = "Element Distinctness"

⟦Similar, but...⟧ (Total.)

Given: $w \in \text{COLORS}^N$. Does w have any repetitions, or all w_i distinct?

$D(\varphi) = N$. $R(\varphi) \gtrsim N$ ⟦Imagine all w_i distinct except 2. Have to find them...⟧

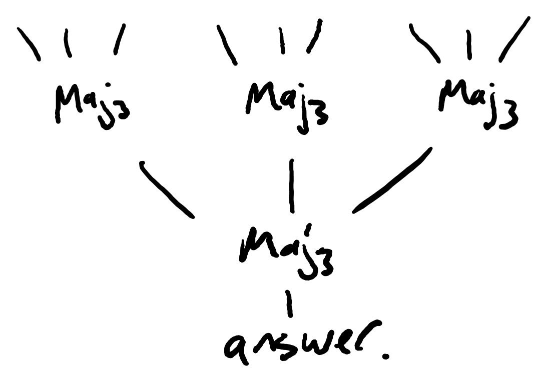
$Q(\varphi) \lesssim N^{3/4}$ (not hard, Grover tricks)

$\lesssim N^{2/3}$ (hard - Ambainis '07)

$\gtrsim N^{2/3}$ (follows from $\gtrsim N^{1/3}$ for 2-to-1).

φ = Recursive-Maj_3. COLORS = $\{0,1\}$.

w = 1 0 1 0 0 0 0 1 0



height h, $N := 3^h$.

$D(\varphi) = 3^h = N$. $2.59^h \leq R(\varphi) \leq 2.65^h \leq \left(\frac{8}{3}\right)^h$

[2016] $\downarrow N^{.866}$ [2015] $\downarrow N^{.887}$ exercise

$Q(\varphi) \approx 2^h$ [RS'08]

$= N^{\log_3 2} = N^{.63}$

Remark: We only saw an exponential gap between quantum & classical for promise problems [and not always then].

[Not a coincidence!]

Theorem: [BBCMW'98] $\forall$ total $\varphi : \{0,1\}^N \to \{0,1\}$
$w \mapsto$ Yes/No

$$D(\varphi) \lesssim Q(\varphi)^6.$$

[At best, quantum can give a 1/6-power speedup over classical — even just deterministic!]

Philosophical import: For "unstructured" (= total, non-promise) problems, quantum cannot give exponential speedup — just polynom.

Thm is consequence of:

1. Known classical query stuff from 1989
2. The lower bound $Q(OR_N) \gtrsim \sqrt{N}$.

[Which, again, we'll see next time.]

[[Remember in Lecture 2 I spent a while comparing the mighty magic of Quantum — studied in '90s, '00s — to the mighty magic of Randomness — studied in '70s, '80s? Well, back in the '80s, they were super-curious about $R(\varphi)$ vs. $D(\varphi)$ ...]]

$R(\varphi)$ vs. $D(\varphi)$:

For <u>partial</u> φ, enormous speedups poss.

E.g.: $w \in \{0,1\}^N$ is either $w = 00\cdots0$ or w has $N/2$ 1's. Decide which.

$D(\varphi) = \frac{N}{2} + 1$. $R(\varphi) = 2$. (!)

For <u>total</u> φ [[Hmm. We didn't see <u>any</u> difference until Rec-Maj3]]

[Nisan '89] <u>Theorem</u>: $D(\varphi) \lesssim R(\varphi)^3$.

<u>Philosophical Import:</u> same (?!)

Nisan's Proof:

0. Invented a new complexity measure:
   Embedded OR complexity$(\varphi) \equiv EOC(\varphi)$
   ⟦Well, that's just my pet name for it.⟧
   (Real name: "Block Sensitivity (φ)" $\equiv bs(\varphi)$.)

⊛ 1. Showed $D(\varphi) \lesssim EOC(\varphi)^4$.

2. $R(\varphi) \gtrsim EOC(\varphi)$ is a trivial consequence of $R(OR_N) \gtrsim N$.

   ⟦Because the def^n of $EOC(\varphi) = M$ is basically "φ hides a copy of the OR function on M bits", as we shall see.⟧

$\therefore\ D(\varphi) \lesssim R(\varphi)^4$. → 3 by minor trick. Never mind.

But: [BBBV'95]: $Q(OR) \gtrsim \sqrt{N} \overset{\text{trivial}}{\Longrightarrow} Q(\varphi) \gtrsim \sqrt{EOC(\varphi)}$

$\Rightarrow EOC(\varphi) \lesssim Q(\varphi)^2$.

$\therefore\ D(\varphi) \lesssim EOC(\varphi)^4 \lesssim Q(\varphi)^8$. → 6 by minor trick – never mind.

Q1: How to show $D(\varphi) \lesssim EOC(\varphi)^4$?

A: Elementary 1-page – or 6-part homework exercise – argument. ☺

Q2: What is $EOC(\varphi)$?

A: Given $\varphi : \{0,1\}^N \to \{0,1\}$, consider a game.

You want to reduce φ to an OR function on M bits, M as big as possible, via these moves:

① Fix some bits of w to 0 or 1.

⟦This decreases input length.⟧

Like $\varphi \rightsquigarrow \varphi(1, w_2, w_3, 0, w_5, w_6, \dots, w_N)$

② Negate φ $(\varphi \rightsquigarrow \neg\varphi)$

③ Negate sense of some inputs.

Like $\rightsquigarrow \varphi(w_1, \neg w_2, \neg w_3, w_4, \dots, \neg w_n)$

④ Identify some vbls.

Like $\rightsquigarrow \varphi(w_1, w_2, w_1, w_4, w_1, w_1, w_7, \dots)$

⟦Also decreases input length.⟧

$EOC(\varphi)$ = largest M such that you can produce OR_M in this way.

E.g.: • $\varphi = AND_N$. $EOC(\varphi) = N$.

↳ negate φ & neg. all inputs.

• $\varphi(w) = (w_1 \wedge w_2) \vee (w_3 \wedge w_4) \vee (w_5 \wedge w_6) \vee \cdots$

$EOC = N/2 \rightarrow$ identify w_1, w_2
w_3, w_4
etc.

<u>Notice</u>: Rules ①, ②, ③, ④ can only make query complexity of φ (D/R/Q) smaller, not larger. [[Actually, ②, ③ leave it unchanged.]]

$$\Rightarrow \quad D(\varphi) \gtrsim D(OR_{EOC(\varphi)}) = EOC(\varphi)$$
$$R(\varphi) \gtrsim R(OR_{EOC(\varphi)}) \gtrsim EOC(\varphi)$$
$$Q(\varphi) \gtrsim Q(OR_{EOC(\varphi)}) \gtrsim \sqrt{EOC(\varphi)}.$$

Formal Definition:

"EOC(φ)" = block-sensitivity(φ) =

maximal M such that there exists:

(i) $w \in \{0,1\}^N$

(ii) disjoint sets of coords. $J_1, \ldots, J_M \subseteq [N]$

such that $\varphi(w) \neq \varphi(w^{\oplus J_i}) \ \forall i = 1 \ldots M$,

where $w^{\oplus J_i}$ denotes w with all bits in J_i negated.

A few more known facts:

a) $\exists$ partial φ with $Q(\varphi)=1$, $R(\varphi) \gtrsim \frac{\sqrt{N}}{\log N}$

[AA14]

b) Conjectured $\exists$ partial φ with $Q(\varphi) \leq \text{polylog} N$,

$R(\varphi) \geq \frac{N}{\text{polylog} N}$

c) $\exists$ total φ with $Q(\varphi)^{2.5} \lesssim R(\varphi)$

$\rightarrow$ "super-Grover separation" [Ben-David '15]

$\rightarrow$ would be $Q(\varphi)^3 \lesssim R(\varphi)$ assuming (b)