Lecture 10

Even within the query model, it is unsatisfactory that the exponential quantum speedup for the DJ problem only holds when we demand certain correctness. This raises a natural question:

> Can we have an exponential speedup in the query model if we don't demand certain correctness, but say 99.99% correctness?

It turns out the answer is yes, as can be witnessed by Simon's problem. This problem inspired Shor's algorithm, which in some sense instantiates the given function in Simon's problem as a specific circuit yet the exponential speedup persists as far as we know.

Simon's problem

Definition 3 (Simon's problem)**.** For $n \in \mathbb{N}$, define the set of functions:

$$D_0 = \left\{ f : \{0,1\}^n \to \{0,1\}^n \;\middle|\; f \text{ is a bijection} \right\},$$
$$D_1 = \left\{ f : \{0,1\}^n \to \{0,1\}^n \;\middle|\; \text{there exists unique } s \neq 0^n \text{ such that for all } x, y \in \{0,1\}^n \colon f(x) = f(y) \iff x \in \{y, y \oplus s\} \right\}.$$

Problem: given query access to $f \in D_0 \cup D_1$, determine whether $f \in D_0$ or $f \in D_1$.

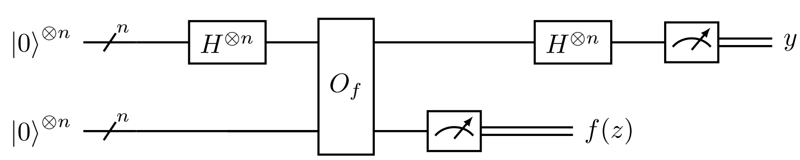
Functions $f \in D_1$ are 2-to-1, i.e., every image of f has exactly two preimages. But not all 2-to-1 functions are in D_1 (why?). The s corresponding to an $f \in D_1$ is known as the period of f.

Classical query complexity. For deterministic computation, it is not too hard to see query complexity is $\leq 2^{n-1} + 1$. But, in fact, the tight bound is $\Theta(2^{n/2})$ like in the randomized case. For the upper bound, the idea is sort of like a "derandomized" version of the randomized algorithm below. For more, see John Watrous's answer to this [StackExchange post]. For randomized computation, the query complexity is $\Theta(2^{n/2})$.

1. Upper bound. Consider querying the value of f on a random subset of M points. The probability that a pair of distinct points map to the same value under f is $1/(2^n - 1) \approx 1/2^n$. So if we know the value of f on $\approx 2^n$ pairs then we can get the probability close to $2^n \times 1/2^n = 1$.[3] But to get the value of f on 2^n pairs, only need to query f on $M \approx \sqrt{2} \times 2^{n/2}$ points so that $\binom{M}{2} \approx M^2/2 \approx 2^n$. This is basically the birthday paradox handwave argument. It can be made rigorous, see, e.g., Proposition 7 in these notes of mine (the n there corresponds to 2^n here).

2. Lower bound. The intuition is that any pair of inputs mapping to distinct values *only* rules out one s so need to query f on at least $\Omega(2^{n/2})$ inputs to rule out all possible s. Will do this more rigorously next lecture.

Quantum query complexity. The quantum algorithm works by repeating the following circuit $O(n)$ times and doing classical post-processing. It solves Simon's problem using $O(n)$ queries, which is exponentially smaller than the randomzied query complexity of $\Theta(2^{n/2})$!



Analysis:

1. Initialize with $|0^n\rangle |0^n\rangle$.

2. Apply $H^{\otimes n}$ to the first register (i.e., first n qubits).[4]

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |0^n\rangle \tag{10}$$

3. Apply O_f: $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$ to obtain

$$\frac{1}{\sqrt{2^n}} \sum_{x} |x\rangle |f(x)\rangle \tag{11}$$

[3] This is a hand wave as probability is not additive like this, more precisely $\Pr[\cup A_i] \neq \sum_i \Pr[A_i]$ in general.

[4] The word "register" refers to a collection of qubits. I'm choosing to refer to the first n qubits as the "first register" here for convenience.

4. Measure the second register (i.e., last n qubits), suppose outcome is $f(z)$ for some $z \in \{0,1\}^n$.
   If $f \in D_0$, then the state of the first register collapses to $|z\rangle$.
   If $f \in D_1$ and the period of f is s, then the state of the first register collapses to
$$\frac{1}{\sqrt{2}}\left(|z\rangle + |z \oplus s\rangle\right). \tag{12}$$
5. Apply $H^{\otimes n}$ to the first register and measure all n qubits.
   If $f \in D_1$: the state becomes
$$\frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} \left((-1)^{z\cdot y} + (-1)^{(z\oplus s)\cdot y}\right)|y\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{\substack{y\in\{0,1\}^n \\ y\cdot s = 0}} (-1)^{z\cdot y}|y\rangle\,; \tag{13}$$
   upon measurement, the output y is a uniformly random element in $\{x \in \{0,1\}^n : x \cdot s = 0 \mod 2\}$, which has size 2^{n-1} (why?).
   If $f \in D_0$: the state becomes
$$\frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{y\cdot z}|y\rangle\,; \tag{14}$$
   upon measurement, the output y is a uniformly random element in $\{0,1\}^n$.
6. Repeat these steps $K = O(n)$ (the precise setting of K depends on the desired success probability, see what follows) times and collect the ys into the rows of a $K \times n$ matrix $A \in \mathbb{F}_2^{K\times n}$. Output D_0 if A has rank n and D_1 if A has rank less than n, where the rank is defined over $\mathbb{F}_2$. (Note that the rank of A is at most n.)

Query complexity. Equal to $K = O(n)$ by definition since each repeat uses only 1 query.

Correctness. In the following, all linear-algebraic notions (such as rank) are with respect to the field $\mathbb{F}_2$.[5] First note that if $f \in D_1$ then A must have rank less than n since the rows of A are all orthogonal to s.[6] Therefore, it suffices to lower bound the probability that the rank of A is equal to n as a function of K, which is done by the following lemma.

Lemma 2. *Let $K \in \mathbb{N}$. Suppose $y_1, \ldots, y_K$ are iid chosen uniformly at random from $\mathbb{F}_2^n$. Then*
$$\Pr[\mathrm{rk}(A) = n] \geq 1 - 2^{n-K}. \tag{15}$$

Proof. Proof based on [StackExchange post]. Since the y_is are chosen uniformly from $\mathbb{F}_2^n$, A is a uniformly random matrix in $\mathbb{F}_2^{K\times n}$. In the following, the probability is over $A \leftarrow \mathbb{F}_2^{K\times n}$.

$$\begin{aligned}
\Pr[\ker(A) \neq \{0\}] &= \Pr[\exists x \in \mathbb{F}_2^n \setminus \{0\}, Ax = 0] && \text{definition} \\
&\leq \sum_{x \in \mathbb{F}_2^n\setminus\{0\}} \Pr[Ax = 0] && \text{union bound} \\
&= (2^n - 1)\frac{1}{2^K} && x \neq 0 \implies Ax \text{ is uniformly random in } \mathbb{F}_2^K \text{ (exercise)} \\
&\leq \frac{2^n}{2^K}.
\end{aligned}$$

Therefore, $\Pr[\mathrm{rk}(A) = n] = \Pr[\ker(A) = \{0\}] \geq 1 - 2^{n-K}$, where the first equality follows from the rank-nullity theorem. □

Therefore, by choosing $K = n + 100$, say, the probability that the rank of A is equal to n is at least $1 - 2^{n-K} \geq 1 - 2^{-100}$, which is very close to 1.

Comment: Pierre asked the question whether the quantum algorithm can be made to succeed with probability 1 (without much blow up in complexity). In class, I answered no. That's the wrong answer, sorry! Apparently, there's an old paper addressing precisely this question, showing that the answer is yes!

[5] Almost all linear-algebraic results you've seen over the reals (such as the rank-nullity theorem) also hold over finite fields.

[6] A rigorous proof: note that $As = 0$ and $s \neq 0$ so $n(A) > 0$, so the rank-nullity theorem, i.e., $\mathrm{rk}(A) + n(A) = n$, implies $\mathrm{rk}(A) < n$.