So far, our discussion of quantum algorithms has focused on what quantum computer's can do: what problems can they solve faster than classical computers -- and how much faster? It is also valuable to ask -- what can't quantum computers do? Can we find lower bounds on the time or number of queries needed to solve certain problems?

We saw in the previous lecture that a quantum computer can do exhaustive search with a quadratic speedup relative to a classical computer. We can also derive lower bounds on the number of queries needed to search or to solve other oracle problems. This provides a more precise characterization of the hardness of these problems.

For discussing upper and lower bounds on complexity, it is useful to use the notation: O ("Big Oh"), Ω ("Big Omega"), Θ ("Big Theta"). We use Big Oh when describing upper bounds on complexity, Big Omega when describing lower bounds, and Big Theta when the upper and lower bounds scale the same way (in which case we say the bounds are "tight"). That is, suppose A and B are two functions of the input size n. Then

$A = O(B)$ means A grows no faster than B:

$A(n) \le \text{const} \times B(n)$ for n sufficiently large

(B is an "upper bound" on A)

– $A = \Omega(B)$ means $B = O(A)$, or in other words B grows no faster than A:

$A(n) \geq \text{const} \times B(n)$ for n sufficiently large

(B is a lower bound on A)

- $A = \Theta(B)$ means $A = O(B)$ and $B = O(A)$:

A and B grow at the same rate as a function of input size n

We derive upper bounds on complexity by discovering algorithms. Grover's algorithm shows that the query complexity of quantum searching for a unique marked state is $O(\sqrt{N})$, where $N = 2^n$ is the size of the set we are searching. Upper bounds are happy news; they characterize things we can do.

But lower bounds are sad news; they limit what we can do. The sad news about quantum searching is that the query complexity is $\Omega(\sqrt{N})$ — we need const. $\times \sqrt{N}$ queries to search with constant probability of success.

On the other hand, complexity theorists are happy when upper and lower bounds "match," since this means we have a precise understanding of the hardness of a problem.

Now -- how do we derive the lower bound on quantum searching? We consider the case where a single string ω is marked by the oracle, so that the unitary applied when we query the box is

$$U_\omega = I - 2|\omega\rangle\langle\omega|$$

A circuit with all together T queries generates a unitary transformation of the form

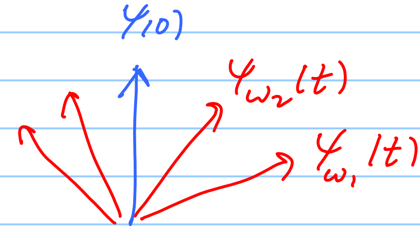
$$U(\omega, T) = U_\omega U_T U_\omega U_{T-1} \cdots U_3 U_\omega U_2 U_\omega U_1 ;$$

The unitaries U_1, U_2, etc are applied in between the queries. These do not depend on the marked string ω, but are otherwise arbitrary. This unitary maps an initial state $|\psi(0)\rangle$ to

$$|\psi_\omega(T)\rangle = U(\omega, T)|\psi(0)\rangle$$

If we fix the circuit, this is one of N possible states (one for each possible value of the marked string ω).

If we are able to identify ω with high success probability, then the states $\{|\psi_\omega(T)\rangle\}$ must be close to an orthonormal basis in the N-dimensional space. As the number t of queries increases, the vectors $|\psi_\omega(t)\rangle$ splay outward, becoming more distinguishable.



But the unitary transformations in between the queries rotate the whole bundle of vectors rigidly without improving their distinguishability. Each query separates the vectors only slightly, so that many queries are needed to achieve high distinguishability.

For keeping track of the splaying of the vectors, we may adopt an "interaction picture" in which the effects of the non-query unitaries are transformed away. Equivalently we can compare $|\psi_\omega(t)\rangle$ to the state that would result if the oracle were "empty" — i.e. in which U_ω is replaced by I. This "reference state" is

$$|\varphi(t)\rangle = \underbrace{U_t U_{t-1} \cdots U_1}_{\text{same non-query unitaries}} \underbrace{|\psi(0)\rangle}_{\text{same initial state}}$$

By how much does $|\psi_\omega(t)\rangle$ deviate from this reference state?

We write

$$|\psi_w(t)\rangle = |\varphi(t)\rangle + |E_w(t)\rangle$$

where $E_w(t)$ is the "error" or deviation from the reference state after t queries. Then

$$|\psi_w(t+1)\rangle = U_w U_{t+1}|\varphi(t)\rangle + U_w U_{t+1}|E_w(t)\rangle$$

and we may write $U_w = I + (U_w - I)$ to obtain

$$|\psi_w(t+1)\rangle = |\varphi(t+1)\rangle + (U_w - I)|\varphi(t+1)\rangle + U_w U_{t+1}|E_w(t)\rangle$$

$$= |\varphi(t+1)\rangle + |E_w(t+1)\rangle$$

where $|E_w(t+1)\rangle = (U_w - I)|\varphi(t+1)\rangle + U_w U_{t+1}|E_w(t)\rangle$

Therefore, we can bound the increase in the size of the deviation resulting from the $(t+1)$st query, using the triangle inequality:

$$\||E_w(t+1)\rangle\| \le \|(U_w - I)|\varphi(t+1)\rangle + \||E_w(t)\rangle\|$$

Since $(U_w - I) = -2|w\rangle\langle w|$, this bound becomes

$$\||E_w(t+1)\rangle\| \le 2|\langle w|\varphi(t+1)\rangle| + \||E_w(t)\rangle\|$$

After a total of T steps, the accumulated deviation satisfies

$$\||E_w(T)\rangle\| \le 2\sum_{t=1}^{T} |\langle w|\varphi(t)\rangle|$$

Now recall that the Cauchy-Schwarz inequality

$|\langle u|v\rangle| \le \|u\|\cdot\|v\|$ implies

$$\sum_{i=1}^{T} |c_i| \le \sqrt{T}\left(\sum_{i=1}^{T} |c_i|^2\right)^{\frac{1}{2}}$$

(we consider $u = (1,1,1,\ldots,1)$, $v = (|c_1|, |c_2|, \ldots, |c_T|)$).

Thus we find

$$\| |E_\omega(T)\rangle \|^2 = \| |\psi_\omega(T)\rangle - |\varphi(T)\rangle \|^2$$
$$\leq 4T \sum_{t=1}^{T} |\langle \omega | \varphi(t)\rangle|^2$$

By summing over ω, and reversing the order of summation, we obtain

$$\sum_{\omega=0}^{N-1} \| |\psi_\omega(T)\rangle - |\varphi(T)\rangle \|^2$$
$$\leq 4T \sum_{t=1}^{T} \sum_{\omega=0}^{N-1} |\langle \omega | \varphi(t)\rangle|^2$$

Since the states $\{|\omega\rangle\}$ are mutually orthogonal, the sum over ω yields the norm squared of the projection of the normalized vector $|\varphi(t)\rangle$ onto the space spanned by the states $\{|\omega\rangle\}$, which is at most 1, and so we have

$$\sum_{\omega=0}^{N-1} \| |\psi_\omega(T)\rangle - |\varphi(T)\rangle \|^2 \leq 4T \sum_{t=1}^{T} (1) = 4T^2$$

If ω can be identified with success probability one after T queries, then the states $\{|\psi_\omega(T)\rangle\}$ must be perfectly distinguishable — i.e. mutually orthogonal. And if $\{|i\rangle\}$ is any orthonormal basis, then

$$\sum_{i=0}^{N-1} \| |i\rangle - |\varphi\rangle \|^2 = \sum_{i=0}^{N-1} \left(2 - 2\,\mathrm{Re}\langle i|\varphi\rangle\right)$$
$$\geq 2N - 2\sum_{i=0}^{N-1} |\langle i|\varphi\rangle| \geq 2N - 2\sqrt{N}$$
$$= 2N\left(1 - N^{-1/2}\right)$$

(using the Cauchy-Schwarz $\neq$ again).

Therefore, to attain success probability one, the required number of queries T satisfies

$$4T^2 \geq 2N(1-N^{-1/2}) \Rightarrow T \geq \sqrt{\frac{N}{2}}\left(1-N^{-1/2}\right)^{1/2}$$

Comparing to $T \approx \frac{\pi}{4}\sqrt{N}$ — the number of queries needed in Grover's algorithm to reach Prob of success ≈ 1 — we see that our upper bound exceeds the lower bound by only about 11% ($\pi/4 \approx .785$ compared to $1/\sqrt{2} \approx .707$). In fact a more careful analysis (keeping track of angular deviation instead of distance — see Dohotaru + Høyer, arXiv: 0810.3647) yields a lower bound showing that the constant $\pi/4$ cannot be improved at all.

What if we demand a success probability of at least $1-\varepsilon$, rather than 1? In that case (after applying one last unitary that aligns $\{|\psi_\omega\rangle\}$ with $\{|\omega\rangle\}$ as closely as possible):

$$|\psi_\omega\rangle = A_\omega |\omega\rangle + B_\omega |\omega^\perp\rangle$$

where $\langle\omega^\perp|\omega\rangle = 0$, $|B_\omega| \leq \sqrt{\varepsilon}$ and $\sqrt{1-\varepsilon} \leq |A_\omega| \leq 1$

therefore, $|\langle\psi_\omega|\varphi\rangle| = |A_\omega\langle\omega|\varphi\rangle + B_\omega\langle\omega^\perp|\varphi\rangle|$

$$\leq |\langle\omega|\varphi\rangle| + \sqrt{\varepsilon}$$

and we have

$$4T^2 \geq \sum_{\omega=0}^{N-1} \| |\psi_\omega\rangle - |\varphi\rangle \|^2 \geq 2N - 2\sum_{\omega=0}^{N-1}|\langle\psi_\omega|\varphi\rangle|$$

$$\geq 2N - 2\sqrt{N} - 2N\sqrt{\varepsilon}$$

or $$T \geq \sqrt{\frac{N}{2}}\left(1-\sqrt{\varepsilon}-N^{-\frac{1}{2}}\right)^{1/2}$$

$$\simeq \sqrt{\frac{N}{2}}\left(1-\frac{\sqrt{\varepsilon}}{2}-\cdots\right) \quad \text{for } N \gg 1 \text{ and } \varepsilon \ll 1$$

we see that $T = \Omega(\sqrt{N})$ for any constant success probability.

Notice that in order for our lower bound to be saturated, the "error terms" $\langle \omega | e(t) \rangle$ must add together coherently, with a common phase. Evidently, this is what is achieved by Grover's algorithm.

We might also be interested in the case where the oracle either marks a unique marked state $|\omega\rangle$ (where $\omega \in \{0, 1, 2, \ldots, N-1\}$) or else it marks no state at all (i.e., is "empty"). And suppose that in that case we are only interested in whether there is a marked state or not — we do not care which state is marked, only whether there is any marked state.

To be able to distinguish the state $|\psi\rangle$ from any of the states $|\psi_\omega\rangle$, we want $|\psi\rangle$ to have a small enough overlap with every $|\psi_\omega\rangle$. Recall from exercise (2.1)d that if we want to distinguish two pure states $|\chi\rangle$ and $|\eta\rangle$, the optimal error probability is

$$\left|p_{error}-\tfrac{1}{2}\right| = \tfrac{1}{2}\sqrt{1-|\langle \chi|\eta\rangle|^2}$$

As we have just seen, if $|\langle \psi_\omega|\psi\rangle|^2 \leq \varepsilon \quad \forall\, \omega$, then

$$4T^2 \geq 2N - 2\sum_{\omega=0}^{N-1} |\langle \psi_\omega|\psi\rangle| \geq 2N\left(1-\sqrt{\varepsilon}\right)$$

$$\Rightarrow \quad T \geq \sqrt{\frac{N}{2}}\left(1-\sqrt{\varepsilon}\right)^{\frac{1}{2}}$$

That is, we require T at least this large if we are to distinguish the empty oracle from the oracle that marks ω for each possible value of ω, with error prob

$|P_{error} - \frac{1}{2}| \geq \frac{1}{2}\sqrt{(1-\varepsilon)}$, or equivalently

if we can successfully identify the empty oracle with

$P_{success} \geq \frac{1}{2} + \delta$ where $\varepsilon = 1 - 4\delta^2$; thus

$$T \geq \sqrt{\frac{N}{2}}\left(1 - \sqrt{1-4\delta^2}\right) = \text{const} \times \sqrt{N}$$

if δ is a nonzero constant

We have found that, at least in the black box model, quantum computers can speed up exhaustive search at best quadratically, not exponentially. What does this imply about the class BQP? Perhaps we may regard it as indirect evidence that NP is not contained in BQP. At any rate, it seems that the sheer magic of quantum superposition does not suffice to achieve exponential speedups; since unstructured search is not strong enough, exponential speedups can be achieved only by exploiting a problem's structure. If there is any structure shared by all problems in NP, it seems to be deeply hidden, as so far we see no glimpse of it (and so, in the hardest instances, exhaustive search is as good as any other method). If such structure exists at all, it would be a surprise if it turned out to be well matched to the advantages of quantum circuits!

More lower bounds

What more can be said about quantum lower bounds in the oracle setting? We consider a box that evaluates a Boolean function

$$f: \{0,1\}^n \longrightarrow \{0,1\}$$

with $N = 2^n$ possible input values. Thus f can be represented as a binary string of length N

$$f \approx X = X_{N-1} X_{N-2} --- X_2 X_1 X_0$$

where $X_i = f(i)$ is the responce to the query $i \in \{0, 1, --, N\}$. Thus there are 2^N possible oracles. Our task is to determine some global property of X; that is to evaluate $F(X)$ where

$$F: \{0,1\}^N \longrightarrow \{0,1\}$$

encodes the answer to a YES/NO question about the oracle.

It is useful to distinguish a "total" function and a partial function. We say that F is "total" if it is defined on every $X \in \{0,1\}^N$. In that case, the oracle could be evaluating any one of the 2^{2^n} Boolean functions with an n-bit input. We say that F is "partial" if it is defined on a restricted domain. In that case, the formulation of the problem includes a <u>promise</u>: the oracle has special properties.

For example, in the case of the Deutsch-Jozsa problem ("constant" vs "balanced"), X has the restricted Hamming weight

$$|X| \in \{0, \tfrac{N}{2}, N\}$$

(The Hamming weight is the number of 1's in the binary string X; for a constant function either $|X| = 0$ or $|X| = N$, while for a balanced function $|X| = N/2$.)

For the case of Grover's search problem with one marked state, $|X|=1$. For the case of Simon's problem, where $f: \{0,1\}^n \to \{0,1\}^n$ and the question is whether the function is 2-1 or 1-1, X actually has length $n2^n$ (i.e., it is really n Boolean functions) and the promise $f(x) = f(x \oplus a)$ for some a restricts the family of strings $\{X\}$ under consideration

Simon's algorithm demonstrates that exponential speedups in oracle complexity are possible in some cases. Yet Grover's search for a unique marked state shows that for some partial functions, an exponential (or even superquadratic) speedup cannot be achieved.

Total functions (problems with no promise) are in some sense harder to evaluate than partial functions, since the oracle X is unrestricted. How hard are they?

Actually, we have already encountered one interesting total function! The OR function

$$OR(X) = \begin{cases} 0 & X = 0 \\ 1 & \text{else} \end{cases}$$

This function answers the question: is any string marked by the oracle? We have already discussed how $OR(X)$ can be evaluated with high success probability in $O(\sqrt{N})$ queries, using Grover's algorithm. In fact, $\Omega(\sqrt{N})$ queries are required, to evaluate OR, since we have shown $\Omega(\sqrt{N})$ are necessary to distinguish the "empty" oracle from one that marks a unique state.

For a function $F(X)$, we may define $Q_2(F)$ — the probabilistic quantum query complexity of F. This is the minimum number of queries needed to output $F(X)$ with success probability $\geq 2/3$. (The subscript 2 indicates that the error is "two-sided" — we are allowed to be wrong with prob $\le \frac{1}{3}$ for either value of $F(X)$.) We may thus say that

$$Q_2(OR) = \Theta(\sqrt{N})$$

Thus, we know that some partial functions admit exponential quantum speedups and that the total function OR admits a quadratic speedup. What about other total functions? In fact, for total functions there are no exponential quantum speedups in query complexity, and the "typical" speedup is just a factor of 2!

To obtain lower bounds on query complexity for total functions, it is useful to apply the "method of polynomials." We note that for $X_i \in \{0,1\}$, $X_i^2 = X_i$. This means that we may regard $F(X)$ as a multilinear polynomial in $\{X_0, X_1, \ldots, X_{N-1}\}$ with degree at most N, and the polynomial of minimal degree expressing $F(X)$ is unique. For example,

$$OR(X) = 1 - (1-X_0)(1-X_1) \cdots (1-X_{N-1}),$$

a polynomial of degree N. Although $\{X_i\}$ are really binary variables, it can be helpful to regard $F(X)$ as a polynomial function mapping $\mathbb{R}^N \to \mathbb{R}$ rather that $\{0,1\}^N \to \{0,1\}$

In a quantum algorithm for evaluating $F(X)$, after T queries we obtain a quantum state $|\psi(X)\rangle$, and then we attempt to read out the value of $F(X)$ by performing a POVM with two outcomes $\{E_0, E_1\}$, where $E_0 + E_1 = I$. The prob. distribution for the outcomes is

$$\text{Prob}(0, X) = \langle \psi(X) | E_0 | \psi(X) \rangle$$
$$\text{Prob}(1, X) = \langle \psi(X) | E_1 | \psi(X) \rangle$$

If our algorithm has success probability $\geq 2/3$, then

$$\text{Prob}(0, X) \geq 2/3 \quad \text{for } F(X) = 0,$$
$$\text{Prob}(1, X) \geq 2/3 \quad \text{for } F(X) = 1.$$

Equivalently,
$$|\text{Prob}(1, X) - F(X)| \leq \frac{1}{3}$$
for all X. In this sense, then, for a successful algorithm, $\text{Prob}(1, X)$ is a good approximation to $F(X)$.

Now, the key point is that the X dependence of $|\psi(X)\rangle$ arises only from the queries, and the number of queries is related to the degree of the polynomial $\text{Prob}(1, X)$. In each query, the oracle applies the unitary

$$U_X : |i, y, z\rangle \longmapsto |i, y \oplus X_i, z\rangle$$

where $i \in \{0, 1, \ldots, N-1\}$ is the query, $X_i = f(i)$ is the oracle's response to the query, and $|z\rangle$ is a basis state for all the work qubits used by the algorithm. Thus, U is a direct sum of 2×2 blocks, where the blocks are labeled by (i, z), and acting within the block

$$U_X^{(i)} = \begin{pmatrix} 1 - X_i & X_i \\ X_i & 1 - X_i \end{pmatrix}.$$

That is $U_X^{(i)}$ is the identity for $X_i = 0$, and a bit flip for $X_i = 1$. What is important is that U_X is linear in X, which means that after T queries $|\psi(X)\rangle$ can be expressed as a polynomial in X of degree at most T; therefore

$$\mathrm{Prob}(1,X) = \langle \psi(X) | E_1 | \psi(X) \rangle$$

is a polynomial in X of degree at most $2T$!

We conclude:

$F(X)$ can be computed, with success probability $\geq 2/3$ for each input X, after T queries only if $F(X)$ can be approximated by a polynomial of degree $\leq 2T$.

Therefore, if $\widetilde{\deg}(F)$ denotes the degree of the minimal-degree polynomial $P(X)$ such that

$$|P(X) - F(X)| \leq \frac{1}{3} \text{ for all } X, \text{ then}$$

$$Q_2(F) \geq \frac{1}{2} \widetilde{\deg}(F)$$

Note that, although the minimal degree polynomial expressing $OR(X)$ exactly has degree N, this argument shows that $OR(X)$ can be approximated by a polynomial of degree $O(\sqrt{N})$.

Another example of a total function is

$$PARITY(X) = \begin{cases} 0 & \text{if } |X| = \text{even} \\ 1 & \text{if } |X| = \text{odd} \end{cases}$$

where $|X|$ denotes the Hamming weight of the string X. Classically, it is evident that N queries are necessary to evaluate PARITY, since any one of the N input bits can flip the value of $F(X)$.

Quantumly, it is easy to see that

$$Q_2(PARITY) \leq N/2.$$

That is because we can use the trick in Deutsch's algorithm to evaluate the parity $X_i \oplus X_j$ of two input bits

If we query the oracle with $\frac{1}{\sqrt{2}}(|i\rangle + |j\rangle)$, then

$$\frac{1}{\sqrt{2}}(|i\rangle + |j\rangle) \longmapsto \frac{1}{\sqrt{2}}\left[(-1)^{X_i}|i\rangle + (-1)^{X_j}|j\rangle\right]$$

Then we can measure in the basis $|\pm\rangle = \frac{1}{\sqrt{2}}(|i\rangle \pm |j\rangle)$ obtaining $|+\rangle$ if $X_i \oplus X_j = 0$ and $|-\rangle$ if $X_i \oplus X_j = 1$. Repeating $N/2$ times for $N/2$ disjoint pairs of input bits, we evaluate

$$\text{PARITY}(X) = X_0 \oplus X_1 \oplus X_2 \oplus X_3 \oplus \cdots \oplus X_{N-2} \oplus X_{N-1}$$

This is a factor of 2 speedup relative to the best classical algorithm.

Furthermore, it turns out that

$$\widetilde{\deg}(\text{PARITY}) = N \quad , \text{ so that}$$

$$Q_2(\text{PARITY}) \geq \frac{1}{2}\widetilde{\deg}(\text{PARITY}) = N/2.$$

The factor of 2 speedup is actually optimal.

How do we see that $\widetilde{\deg}(\text{PARITY}) = N$? First, it is helpful to note that PARITY is a symmetric function — we say F is symmetric if $F(X)$ is invariant under any permutation of its N input bits:

$$F(\pi(X)) = F(X) \quad \text{where } \pi \text{ is a permutation}$$

$$\pi : X_i \longmapsto X_{\pi(i)}, \quad \pi \in S_N$$

We note that if $P(X)$ approximates $F(X)$ where $F(X)$ is symmetric, then the symmetrized $P(X)$

$$P_{sym}(X) = \frac{1}{N!}\sum_{\pi \in S_N} P(\pi(X))$$

also approximates $F(X)$. This is true because

$$\frac{1}{3} \geq |P(\pi(X)) - F(\pi(X))| = |P(\pi(X)) - F(X)|$$

for each permutation π.

Furthermore P_{sym} is a function of the Hamming weight $|X|$ of X; it can be regarded as a polynomial in the single variable $|X|$, whose degree is at most the degree of $P(X)$. We may write

$$P_{sym}(X) = C_0 + C_1 V_1 + C_2 V_2 + \cdots + C_2 V_d$$

where:

V_1 is obtained by symmetrizing a linear term $\Rightarrow$

$$V_1 = X_0 + X_1 + \cdots + X_N = |X|.$$

V_2 is obtained by symmetrizing a quadratic term

$$V_2 = \sum_{i<j} X_i X_j = \binom{|X|}{2}$$

(Recall $X_i^2 = X_i$, and $X_i X_j \neq 0$ only if $X_i = X_j = 1$)

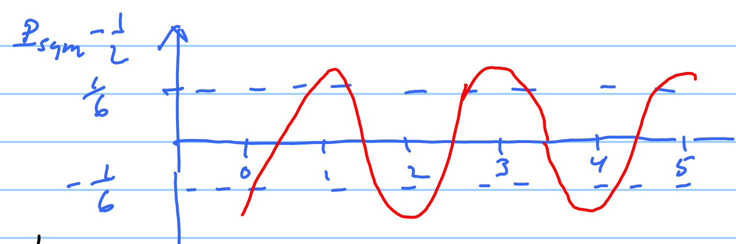
Similarly

$$V_k = \sum_{i_1 < i_2 < \cdots < i_k} X_{i_1} X_{i_2} \cdots X_{i_k} = \binom{|X|}{k} \quad \text{for } k = 3, 4, \ldots, d$$

Therefore, if $P(X)$ approximates $F(X)$, then $P_{sym}(|X|)$ approximates $F(|X|)$, or

$$|P_{sym}(|X|) - F(|X|)| \leq \frac{1}{3} \quad \text{for } |X| = 0, \ldots, N$$

Thus if we plot $P_{sym}(|X|) - \frac{1}{2}$ as a function of the <u>real</u> variable $|X|$, it take value $\leq -\frac{1}{6}$ for even $|X|$ and value $\geq \frac{1}{6}$ for odd $|X|$. Therefore it must cross zero at least N times for $|X| \in (0, N)$



a real polynomial with N real zeros must have degree at least N. Therefore P_{sym} has degree $\geq N$ and so does P. We conclude that

$$\widetilde{\deg}(\mathrm{PARITY}) \geq N \quad \text{and} \quad Q_2(\mathrm{PARITY}) = N/2$$

In fact, "most functions" are like PARITY: the quantum speedup is at best a constant factor. Consider for example a random symmetric function $F(|X|)$. Each time $|X|$ advances by 1, the value of F changes with probability $\frac{1}{2}$, so that the approximating polynomial $P_{sym}(|X|) - \frac{1}{2}$ crosses zero with prob $\frac{1}{2}$. Therefore typical symmetric polynomial has approximating polynomial with degree linear in N, and hence $Q_2(F) = \Omega(N)$.

Now, we don't necessarily care much about "most functions." But in fact it can be shown that for any nonconstant symmetric total function,

$$\widetilde{\deg}(F) = \Omega(\sqrt{N}) \Rightarrow Q_2(F) = \Omega(\sqrt{N})$$

– a quadratic speedup is the best possible and the OR function demonstrates that this is tight. In a sense, of all nontrivial, symmetric total functions, OR is the "easiest" to approximate by a polynomial $P_{sym}(|X|)$.

The polynomial method also yields lower bounds on total functions that are not symmetric and symmetric functions that are not total. For example, let $D(F)$ denote the deterministic classical query complexity of F – the minimum number of queries needed to determine $F(X)$ with certainty for any X. It has been shown that

$$D(F) \leq 216\,[\widetilde{\deg}(F)]^6,$$

and since $Q_2(F) \geq \frac{1}{2}\widetilde{\deg}(F)$ we conclude that

$$Q_2(F) = \Omega\left[(D(F))^{1/6}\right].$$

Thus, in the oracle model, there are no exponential quantum speedups for total functions.
The best possible speedup is a 6th power speedup and the best known speedup is a quadratic speedup. Other results rule out exponential speedups for symmetric functions that are not necessarily total.