Quantum query complexity of determining whether a graph is connected

Abbas Mehrabian

University of Waterloo

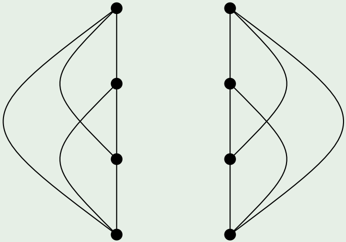
28 November 2013

The connectivity problem

Design an algorithm that determines if a given graph is connected.

Design an algorithm that determines if a given graph is connected.

Example

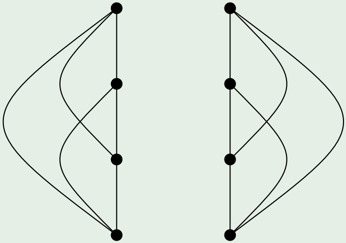


NO

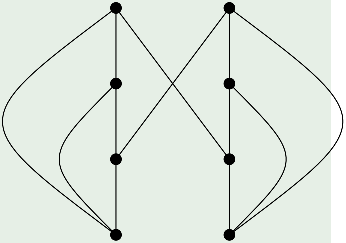
The connectivity problem

Design an algorithm that determines if a given graph is connected.

Example

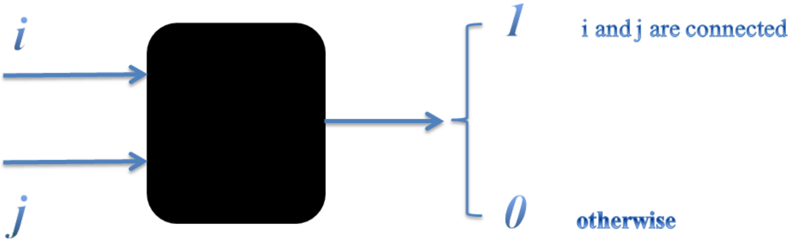


NO



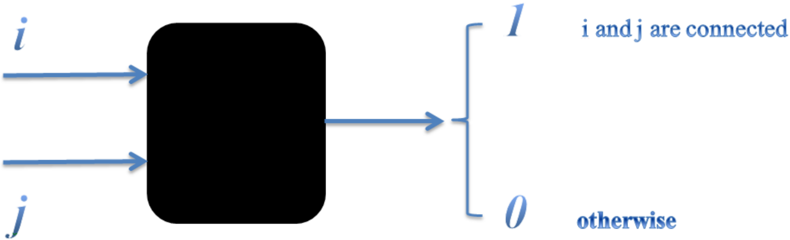
YES

Assume the vertices are numbered from 1 to n.



Access to the input graph

Assume the vertices are numbered from 1 to n.



Each question asked from the black-box is called a query.

Theorem

There is a quantum algorithm with query complexity $O(n\sqrt{n})$ for the connectivity problem with error probability $< 1/3$.

Indeed, this is optimal up to constant factors.

Main results

Theorem

There is a quantum algorithm with query complexity $O(n\sqrt{n})$ for the connectivity problem with error probability $< 1/3$.

Indeed, this is optimal up to constant factors.

Theorem

Every classical algorithm for the connectivity problem that has error probability $< 1/3$ has query complexity $\Omega(n^2)$.

The Quantum Algorithm

Generalization of Grover's search algorithm

Theorem (Grover's search algorithm)

There is a quantum algorithm that given a domain of size N and black-box access to some function $F : \{1, \ldots, N\} \to \{0, 1\}$, for which it is guaranteed that there exists a solution y with $F(y) = 1$, finds a solution evaluating F at $O(\sqrt{N})$ points.

Theorem (Grover's search algorithm)

There is a quantum algorithm that given a domain of size N *and black-box access to some function* $F : \{1, \dots, N\} \to \{0, 1\}$*, for which it is guaranteed that there exists a solution* y *with* $F(y) = 1$*, finds a solution evaluating* F *at* $O(\sqrt{N})$ *points.*

Theorem (Generalization of Grover's search algorithm)

There is a quantum algorithm that if there are s *solutions,*

- ✓ *If* $s > 0$*, outputs a random solution asking an expected number of* $O\left(\sqrt{N/s}\right)$ *queries.*
- ✓ *If* $s = 0$*, the algorithm does not halt.*

Note: the algorithm need not know s*.*

Theorem (Grover's search algorithm)

There is a quantum algorithm that given a domain of size N and black-box access to some function $F : \{1, \dots, N\} \to \{0, 1\}$, for which it is guaranteed that there exists a solution y with $F(y) = 1$, finds a solution evaluating F at $O(\sqrt{N})$ points.

Theorem (Generalization of Grover's search algorithm)

There is a quantum algorithm that if there are s solutions,

- ✓ *If $s > 0$, outputs a random solution asking an expected number of $O\left(\sqrt{N/s}\right)$ queries.*
- ✓ *If $s = 0$, the algorithm does not halt.*

Note: the algorithm need not know s.

This algorithm will be called the search algorithm.

Dealing with the never-stopping issue

We present an algorithm that given black-box access to G,

- ✓ if G is connected, outputs YES after asking $O(n\sqrt{n})$ queries on average; and
- ✓ does not halt if G is not connected.

Dealing with the never-stopping issue

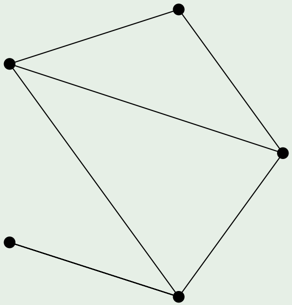
We present an algorithm that given black-box access to G,

- ✓ if G is connected, outputs YES after asking $O(n\sqrt{n})$ queries on average; and
- ✓ does not halt if G is not connected.

It is not hard to turn this into an algorithm with worst-case query complexity $O(n\sqrt{n})$ and error probability $< 1/3$.

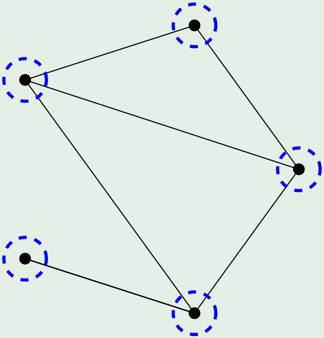
The algorithm

Example



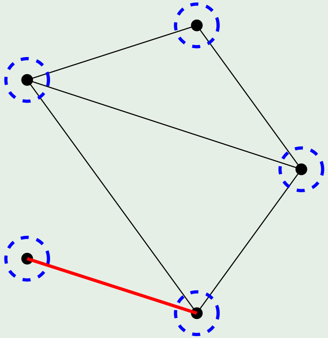
The algorithm

Example

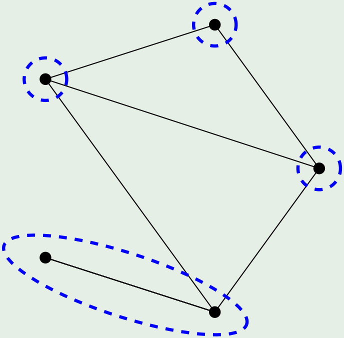


The algorithm

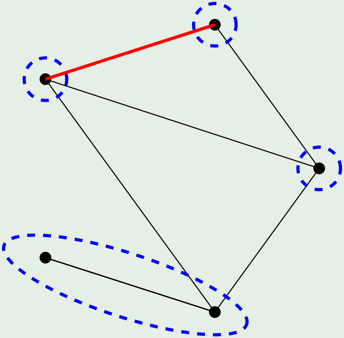
Example



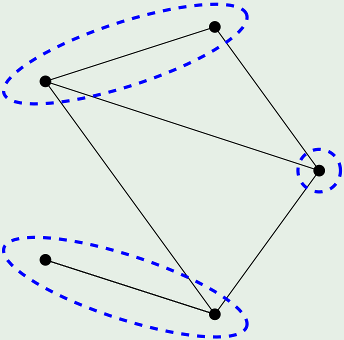
Example



Example

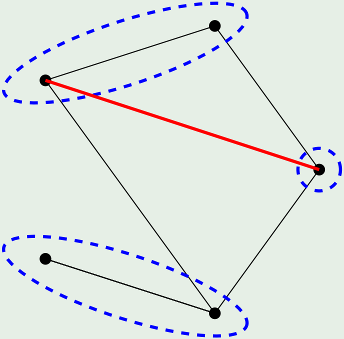


Example



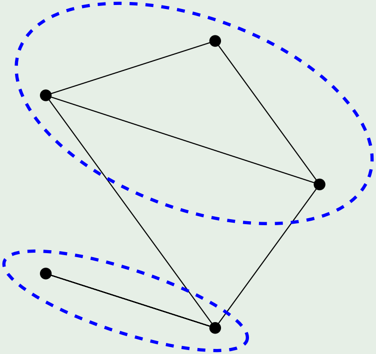
The algorithm

Example



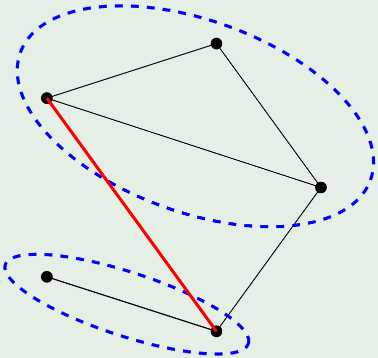
The algorithm

Example



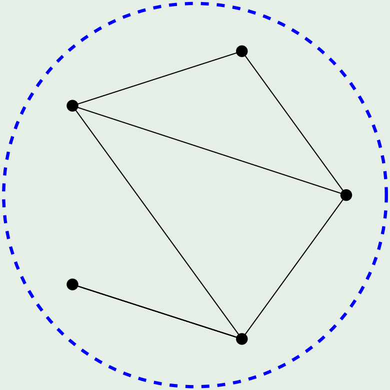
The algorithm

Example

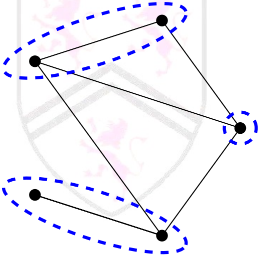


The algorithm

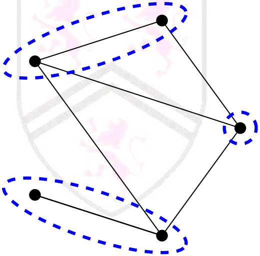
Example



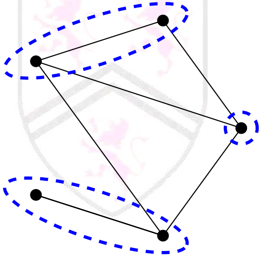
Analysis: connected case



Analysis: connected case



When there are k pieces, there are $\geq k-1$ desirable edges, and domain size is $\binom{n}{2}$. Hence average query complexity of the search algorithm is $O\left(\sqrt{\binom{n}{2} \Big/ (k-1)}\right) = O\left(\sqrt{n^2/k}\right)$.



When there are k pieces, there are $\geq k-1$ desirable edges, and domain size is $\binom{n}{2}$. Hence average query complexity of the search algorithm is $O\left(\sqrt{\binom{n}{2} \Big/ (k-1)}\right) = O\left(\sqrt{n^2/k}\right)$.

Average query complexity of the whole algorithm is

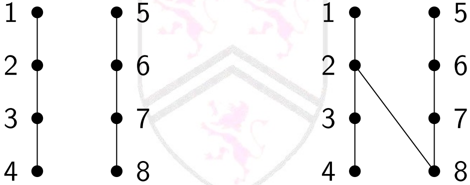
$$\sum_{k=2}^{n} O\left(\sqrt{n^2/k}\right) \leq O\left(n\sqrt{n}\right)$$

Quantum algorithm: the theorem

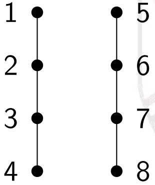
Theorem

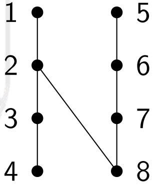
There is a quantum algorithm with query complexity $O(n\sqrt{n})$ *for the connectivity problem with error probability* $< 1/3$.

Classical lower bound



Classical lower bound





$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Classical lower bound: the theorem

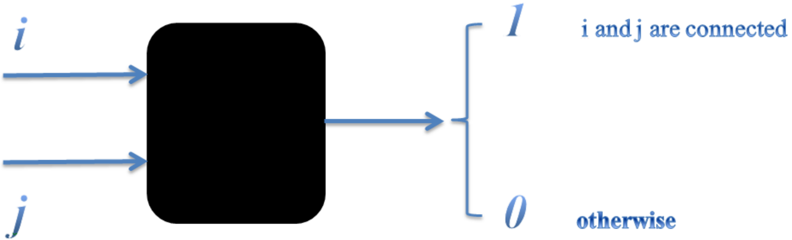
Theorem

Every classical algorithm for the connectivity problem that has error probability $< 1/3$ *has query complexity* $\Omega(n^2)$*.*

Another model

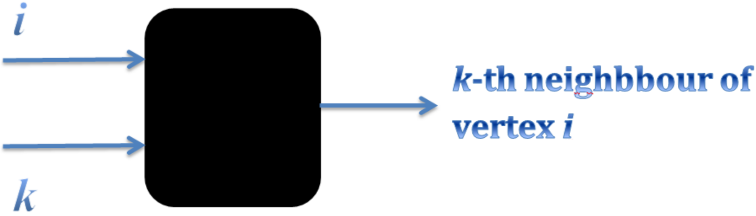
Recall: access to the input graph

Vertices are numbered from 1 to n.



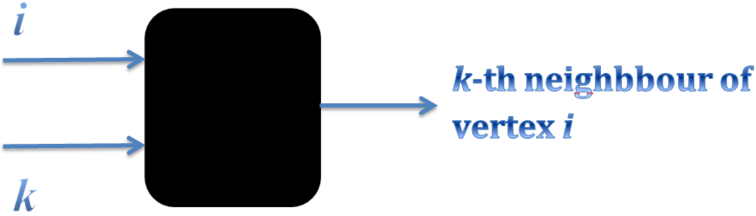
This is called the matrix model.

Vertices are numbered from 1 to n.



A different model

Vertices are numbered from 1 to n.



This is called the array model.

Main results

Theorem

In the array model, there is a quantum algorithm with query complexity $O(n)$ for the connectivity problem with error probability $< 1/3$.

Indeed, this is optimal up to constant factors.

Theorem

In the array model, there is a quantum algorithm with query complexity $O(n)$ for the connectivity problem with error probability $< 1/3$.

Indeed, this is optimal up to constant factors.

Theorem

In the array model, every classical algorithm for the connectivity problem that has error probability $< 1/3$ has query complexity $\Omega(n^2)$.

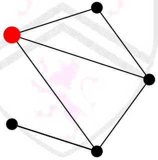
Outline of the previous algorithm

Outline

1. Partition the vertex set into connected pieces.
2. Merge the pieces one by one.

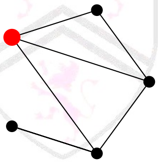
Some definitions

The degree of a vertex is its number of neighbours:

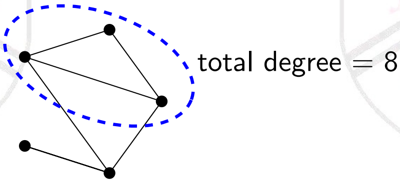


Some definitions

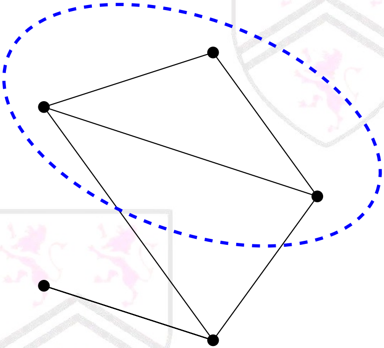
The degree of a vertex is its number of neighbours:



The total degree of a set of vertices S, written $t(S)$, is the sum of degrees of its vertices:

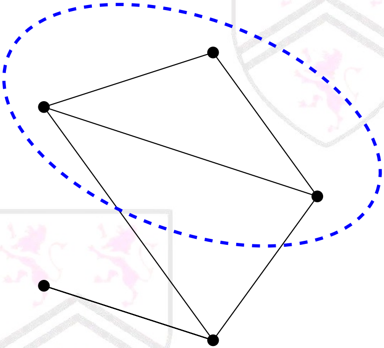


The key observation



total degree $= 8$

The key observation



total degree = 8

In general, we need $O\left(\sqrt{t(P)}\right)$ queries to find an edge going out of P.

The algorithm

In the first phase, asking $O(n)$ classical queries we partition the vertices into connected pieces $P_1, P_2, \ldots, P_k$ such that $t(P_i) < |P_i|^2 \quad \forall i$.

The algorithm

In the first phase, asking $O(n)$ classical queries we partition the vertices into connected pieces $P_1, P_2, \ldots, P_k$ such that $t(P_i) < |P_i|^2 \quad \forall i$.

In the second phase, we merge the pieces iteratively: in every iteration,

1. choose a piece P with minimum total degree.
2. Use the search algorithm to find an edge going out of P.
3. merge two pieces using that edge.

The algorithm

In the first phase, asking $O(n)$ classical queries we partition the vertices into connected pieces $P_1, P_2, \ldots, P_k$ such that $t(P_i) < |P_i|^2 \quad \forall i$.

In the second phase, we merge the pieces iteratively: in every iteration,

1. choose a piece P with minimum total degree.
2. Use the search algorithm to find an edge going out of P.
3. merge two pieces using that edge.

It turns out that if G is connected, then the expected query complexity of the second phase is

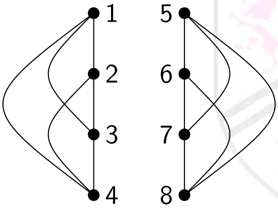
$$O\left(\sqrt{t(P_1)} + \sqrt{t(P_2)} + \cdots + \sqrt{t(P_k)}\right) \leq O(n)$$

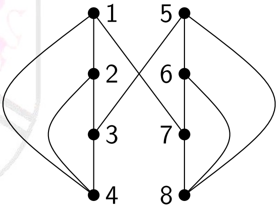
The proved theorem

Theorem

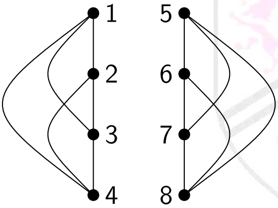
In the array model, there is a quantum algorithm with query complexity $O(n)$ for the connectivity problem with error probability $< 1/3$.

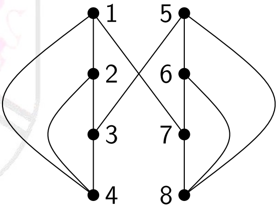
Classical lower bound





Classical lower bound





1	2	4	3
2	1	3	4
3	4	2	1
4	3	1	2
5	7	8	6
6	5	7	8
7	8	5	6
8	7	6	5

1	2	4	7
2	1	3	4
3	4	2	5
4	3	1	2
5	3	8	6
6	5	7	8
7	8	1	6
8	7	6	5

Wrap-up

model	matrix	array
classical	$\Theta(n^2)$	$\Theta(n^2)$
quantum	$\Theta(n\sqrt{n})$	$\Theta(n)$

Wrap-up

model	matrix	array
classical	$\Theta(n^2)$	$\Theta(n^2)$
quantum	$\Theta(n\sqrt{n})$	$\Theta(n)$

Thank you