P versus **NP**

Math 40210, Fall 2015

November 10, 2015

Properties of graphs

A *property* of a graph is anything that can be described without referring to specific vertices

Properties of graphs

A *property* of a graph is anything that can be described without referring to specific vertices

Examples of properties:

- being connected
- having a closed Eulerian trail
- having a Hamiltonian cycle

Properties of graphs

A *property* of a graph is anything that can be described without referring to specific vertices

Examples of properties:

- being connected
- having a closed Eulerian trail
- having a Hamiltonian cycle

Examples of non-properties:

- vertices 1, 2 and 3 form a triangle
- vertices *u* and *v* are in different components

The decision problem for a property

The *decision problem* for a particular property is to determine, for any possible input graph G, whether or not G has the property

The decision problem for a property

The *decision problem* for a particular property is to determine, for any possible input graph G, whether or not G has the property

Examples of decision problems:

- Is G connected?
- Does G have a closed Eulerian trail?
- Is G Hamiltonian (mean, does it have a Hamiltonian cycle)?

The class **P**

A property is *in class* **P** if there is a quick procedure for answering the decision problem, that works for every possible input graph (here "quick" formally means that the number of steps that the procedure takes is at most a polynomial in the number of vertices of the graph)

The class **P**

A property is *in class* **P** if there is a quick procedure for answering the decision problem, that works for every possible input graph (here "quick" formally means that the number of steps that the procedure takes is at most a polynomial in the number of vertices of the graph)

Examples of **P** properties:

- Being connected: form adjacency matrix A, compute $(I + A)^{n-1}$, check if all entries are positive (in which case G is connected), or if there is at least one zero entry (in which case G is not connected)

The class **P**

A property is *in class* **P** if there is a quick procedure for answering the decision problem, that works for every possible input graph (here "quick" formally means that the number of steps that the procedure takes is at most a polynomial in the number of vertices of the graph)

Examples of **P** properties:

- Being connected: form adjacency matrix A, compute $(I + A)^{n-1}$, check if all entries are positive (in which case G is connected), or if there is at least one zero entry (in which case G is not connected)
- Having a closed Eulerian trail: first check that G is connected, then look at the vertex degrees; if they are all even, then G has a closed Eulerian trail; if one or more is odd, then it does not

The class **P**

A property is *in class* **P** if there is a quick procedure for answering the decision problem, that works for every possible input graph (here "quick" formally means that the number of steps that the procedure takes is at most a polynomial in the number of vertices of the graph)

Examples of **P** properties:

- Being connected: form adjacency matrix A, compute $(I + A)^{n-1}$, check if all entries are positive (in which case G is connected), or if there is at least one zero entry (in which case G is not connected)
- Having a closed Eulerian trail: first check that G is connected, then look at the vertex degrees; if they are all even, then G has a closed Eulerian trail; if one or more is odd, then it does not

"**P**" stands for "polynomial time"

The class **NP**

A property is *in class* **NP** if, for every graph G for which the answer to the decision problem is YES, it is possible to present a quick proof of this fact (here again, “quick” formally means that the number of steps that need to be taken to verify that the proof is correct is at most a polynomial in the number of vertices of the graph)

Important note: the *proof* that G has the property has to be short, but there’s no limit to the amount of time needed to come up with the proof

The class **NP**

A property is *in class* **NP** if, for every graph *G* for which the answer to the decision problem is YES, it is possible to present a quick proof of this fact (here again, “quick” formally means that the number of steps that need to be taken to verify that the proof is correct is at most a polynomial in the number of vertices of the graph)

Important note: the *proof* that *G* has the property has to be short, but there’s no limit to the amount of time needed to come up with the proof

Another way to say this: a property is in class **NP** if whenever a graph *G* has the property, I can quickly convince you that it has the property, *as long as I am given as much time as I need to prepare before beginning to convince you*

The class **NP**

A property is *in class* **NP** if, for every graph G for which the answer to the decision problem is YES, it is possible to present a quick proof of this fact (here again, “quick” formally means that the number of steps that need to be taken to verify that the proof is correct is at most a polynomial in the number of vertices of the graph)

Important note: the *proof* that G has the property has to be short, but there’s no limit to the amount of time needed to come up with the proof

Another way to say this: a property is in class **NP** if whenever a graph G has the property, I can quickly convince you that it has the property, *as long as I am given as much time as I need to prepare before beginning to convince you*

“**NP**” stands for “nondeterministic polynomial time”

Examples of **NP** properties

- Being connected: Before talking to you I find a spanning, and I show it to you. **Or**: while you watch, I run the algorithm that answers the decision problem

Examples of **NP** properties

- Being connected: Before talking to you I find a spanning, and I show it to you. **Or**: while you watch, I run the algorithm that answers the decision problem
- Having a closed Eulerian trail: Before talking to you I find a closed Eulerian trail, and I show it to you. **Or**: while you watch, I run the algorithm that answers the decision problem

Examples of **NP** properties

- Being connected: Before talking to you I find a spanning, and I show it to you. **Or**: while you watch, I run the algorithm that answers the decision problem
- Having a closed Eulerian trail: Before talking to you I find a closed Eulerian trail, and I show it to you. **Or**: while you watch, I run the algorithm that answers the decision problem
- Any **P** property is also **NP**: to convince you that G has the property, I run the (quick) algorithm that answers the decision problem

Examples of **NP** properties

- Being connected: Before talking to you I find a spanning, and I show it to you. **Or**: while you watch, I run the algorithm that answers the decision problem
- Having a closed Eulerian trail: Before talking to you I find a closed Eulerian trail, and I show it to you. **Or**: while you watch, I run the algorithm that answers the decision problem
- Any **P** property is also **NP**: to convince you that G has the property, I run the (quick) algorithm that answers the decision problem
- Having a Hamiltonian cycle: Before talking to you, I find a Hamilton cycle in the graph (this may take a **very** long time); once I have found it, I show it to you, and clearly you can quickly verify that it is indeed a Hamiltonian cycle

The \$1,000,000 question

P: properties for which the decision problem can be quickly solved

NP: properties for which a YES answer to the decision problem can be quickly verified, given enough preparation time

The $1,000,000 question

P: properties for which the decision problem can be quickly solved

NP: properties for which a YES answer to the decision problem can be quickly verified, given enough preparation time

We've seen that $\mathbf{P} \subseteq \mathbf{NP}$. The $1,000,000 question is this:

$$\text{Is } \mathbf{P} = \mathbf{NP}?$$

In other words, is it true that every decision problem for which a YES answer can be quickly verified, can also be quickly solved?

The \$1,000,000 question

P: properties for which the decision problem can be quickly solved

NP: properties for which a YES answer to the decision problem can be quickly verified, given enough preparation time

We've seen that $\mathbf{P} \subseteq \mathbf{NP}$. The \$1,000,000 question is this:

$$\text{Is } \mathbf{P} = \mathbf{NP}?$$

In other words, is it true that every decision problem for which a YES answer can be quickly verified, can also be quickly solved?

A specific example: is there a quick way of deciding whether a given graph has a Hamilton cycle?

The class **NP**-complete

A property is in the class **NP**-complete if a procedure for solving the decision problem for that property can be converted into a procedure for solving the decision problem for any other **NP** property, without any significant slow down

The class **NP**-complete

A property is in the class **NP**-complete if a procedure for solving the decision problem for that property can be converted into a procedure for solving the decision problem for any other **NP** property, without any significant slow down

NP-complete properties are in a sense the "hardest" properties: if you solve the decision problem for any one of them, you've solved the decision problem for all other **NP** properties

The class **NP**-complete

A property is in the class **NP**-complete if a procedure for solving the decision problem for that property can be converted into a procedure for solving the decision problem for any other **NP** property, without any significant slow down

NP-complete properties are in a sense the “hardest” properties: if you solve the decision problem for any one of them, you’ve solved the decision problem for all other **NP** properties

Having a Hamiltonian cycle is known to be an **NP**-complete property. So:

- if you find a quick way to answer the question “does G have a Hamiltonian cycle”, you’ve shown $\mathbf{P} = \mathbf{NP}$
- if you prove that no such quick way exists, you’ve shown $\mathbf{P} \neq \mathbf{NP}$

The class **NP**-complete

A property is in the class **NP**-complete if a procedure for solving the decision problem for that property can be converted into a procedure for solving the decision problem for any other **NP** property, without any significant slow down

NP-complete properties are in a sense the "hardest" properties: if you solve the decision problem for any one of them, you've solved the decision problem for all other **NP** properties

Having a Hamiltonian cycle is known to be an **NP**-complete property. So:

- if you find a quick way to answer the question "does G have a Hamiltonian cycle", you've shown $\mathbf{P} = \mathbf{NP}$
- if you prove that no such quick way exists, you've shown $\mathbf{P} \neq \mathbf{NP}$

Computers and Intractability: A Guide to the Theory of NP-Completeness by Garey and Johnson (1979, W. H. Freeman publisher) — 350 pages of **NP**-complete problems

The Millennium Prize Problems

In 2000, the Clay Mathematics Institute identified seven important open problems in mathematics, and offered a prize of $ 1,000,000 for a solution to each one; see http://www.claymath.org/millennium/. One of these seven is the **P** versus **NP** problem

The Millennium Prize Problems

In 2000, the Clay Mathematics Institute identified seven important open problems in mathematics, and offered a prize of $ 1,000,000 for a solution to each one; see http://www.claymath.org/millennium/. One of these seven is the **P** versus **NP** problem

The other six are problems of the form “prove X”, with the $1,000,000 only being offered for a proof of X, not a counterexample. For **P** versus **NP**, the full prize is guaranteed, whichever way the problem is resolved