

CSE 202: Intractability

T.C. Hu

Spring 2007

Time to solve a problem depends on inputs of the problem.

- Length of input
- Instances of input

For example, if n denotes length of input:

Time complexity: $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$

A problem is solved “efficiently” if there exists an algorithm with time complexity that is a polynomial of n . (e.g. $O(n^3)$)

We have solved lots of problems efficiently but also failed many times.

Are problems intrinsically hard or are we stupid?

There exists a class of problems called **NP-complete** problems.

These problems are equivalent, in the sense that if any one of these problems can be solved in polynomial time, then all of them can be solved in polynomial time.

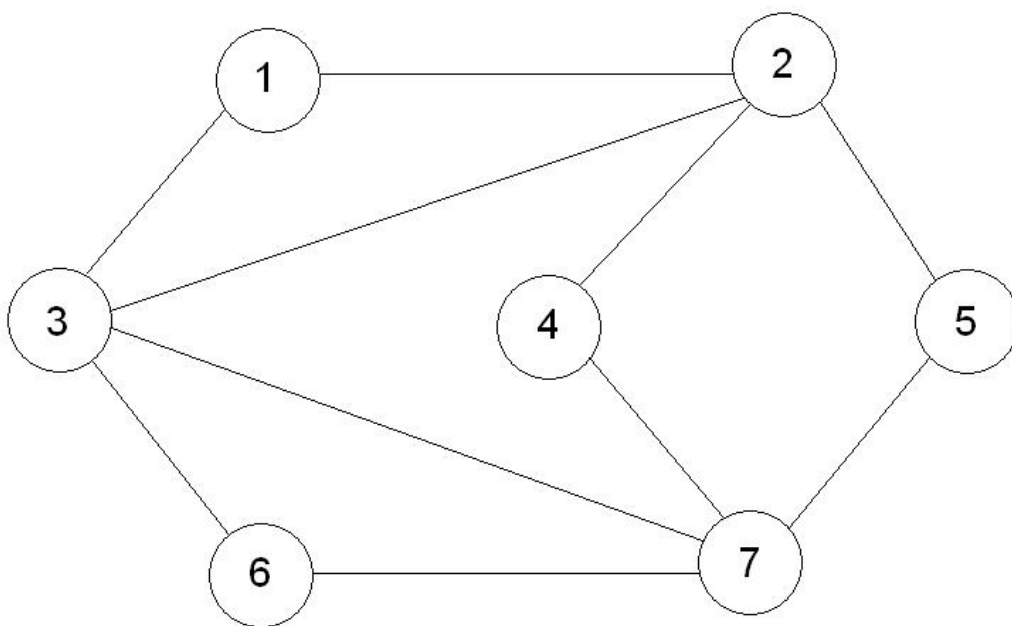
A special case of an NP-complete problem may be solved efficiently.

The following statements are equivalent:

- Problem X is at least as hard as Y
- $Y \leq_p X$
- Y is polynomially reducible to X

If X can be solved in polynomial time, then Y can be solved in polynomial time.

If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time either.



A set of nodes in a graph is **independent** if no two nodes in the set are joined by an edge.

A set of nodes in a graph is a vertex cover if every edge in the graph is incident to some node in the set.