Chapter 6: P vs NP

§6.1 Introduction

An algorithm has *polynomial running time* if there is a constant $c \geq 1$ such that for every input of length n, the algorithm takes $O(n^c)$ time.

§6.1 Introduction

An algorithm has *polynomial running time* if there is a constant $c \geq 1$ such that for every input of length n, the algorithm takes $O(n^c)$ time.

An algorithm has *exponential running time* if there is a constant $c \geq 1$ such that for every input of length n, the algorithm takes $O\left(2^{n^c}\right)$ time.

§6.1 Introduction

An algorithm has *polynomial running time* if there is a constant $c \geq 1$ such that for every input of length n, the algorithm takes $O(n^c)$ time.

An algorithm has *exponential running time* if there is a constant $c \geq 1$ such that for every input of length n, the algorithm takes $O\left(2^{n^c}\right)$ time.

Intuitively speaking,

polynomial means good, fast, efficient, easy, ...

exponential means bad, slow, "try all possible solutions", difficult, ...

Complexity Class P

A *decision problem* is a problem for which the answer is "yes" or "no".

$P =$ set of all decision problems that can be solved in polynomial time

Complexity Class P

A *decision problem* is a problem for which the answer is "yes" or "no".

$$P = \text{set of all decision problems that can be solved in polynomial time}$$

Examples of problems that are in P:

- Is a given input graph connected?

Complexity Class P

A *decision problem* is a problem for which the answer is "yes" or "no".

$$P = \text{set of all decision problems that can be solved in polynomial time}$$

Examples of problems that are in P:

- Is a given input graph connected?
- Is a given input graph bipartite?

Complexity Class *P*

A *decision problem* is a problem for which the answer is "yes" or "no".

$P =$ set of all decision problems that can be solved in polynomial time

Examples of problems that are in P:

- Is a given input graph connected?
- Is a given input graph bipartite?
- Is a given input sequence sorted?

Complexity Class P

A *decision problem* is a problem for which the answer is "yes" or "no".

$P =$ set of all decision problems that can be solved in polynomial time

Examples of problems that are in P:

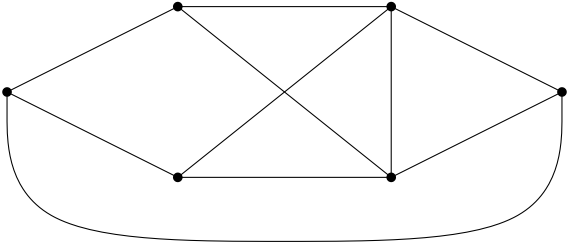
- Is a given input graph connected?
- Is a given input graph bipartite?
- Is a given input sequence sorted?
- Does a given input graph contain an Euler cycle? An *Euler cycle* is a cycle that traverses each edge exactly once.

Other Problems

HAM-CYCLE

input: An undirected graph $G = (V, E)$ stored using adjacency lists.

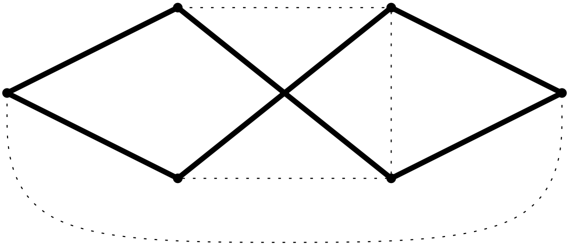
question: Does G contain a Hamiltonian cycle? A *Hamiltonian cycle* is a cycle that traverses each vertex exactly once.



Other Problems
HAM-CYCLE

input: An undirected graph $G = (V, E)$ stored using adjacency lists.

question: Does G contain a Hamiltonian cycle? A *Hamiltonian cycle* is a cycle that traverses each vertex exactly once.

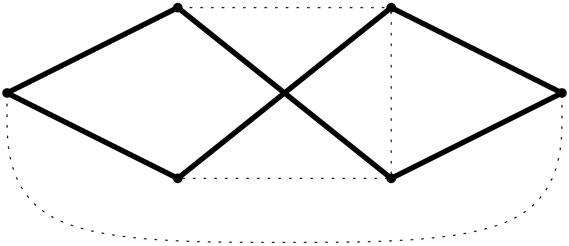


Other Problems

HAM-CYCLE

input: An undirected graph $G = (V, E)$ stored using adjacency lists.

question: Does G contain a Hamiltonian cycle? A *Hamiltonian cycle* is a cycle that traverses each vertex exactly once.



Not known if this problem is in P!

Other Problems

Traveling Salesman Problem (TSP)

input:

- A complete directed graph $G = (V, E)$, where each edge $(u, v) \in E$ has a weight $wt(u, v) > 0$.
- An integer K.

question: Does G contain a Hamiltonian cycle of total weight at most K?

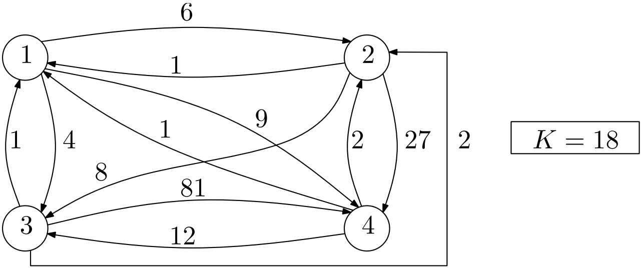
Other Problems

Traveling Salesman Problem (TSP)

input:

- A complete directed graph $G = (V, E)$, where each edge $(u, v) \in E$ has a weight $wt(u, v) > 0$.
- An integer K.

question: Does G contain a Hamiltonian cycle of total weight at most K?



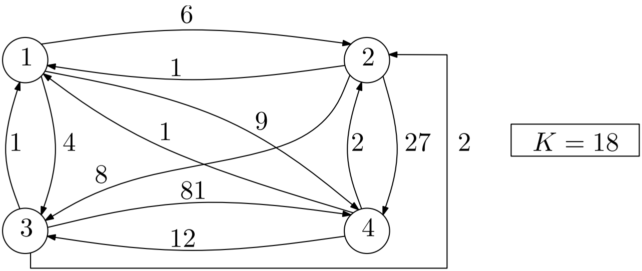
Other Problems

Traveling Salesman Problem (TSP)

input:

- A complete directed graph $G = (V, E)$, where each edge $(u, v) \in E$ has a weight $wt(u, v) > 0$.
- An integer K.

question: Does G contain a Hamiltonian cycle of total weight at most K?



Not known if this problem is in P!

Other Problems

SUBSET-SUM

input: A set S of integers together with an integer t

question: Is there a subset S' of S such that

$$\sum_{x \in S'} x = t?$$

Other Problems

SUBSET-SUM

input: A set S of integers together with an integer t

question: Is there a subset S' of S such that

$$\sum_{x \in S'} x = t?$$

Example:

$$S = \{5, 7, 56, 23, 2902\}$$
$$t = 2963$$

Other Problems

SUBSET-SUM

input: A set S of integers together with an integer t

question: Is there a subset S' of S such that

$$\sum_{x \in S'} x = t?$$

Example:

$$S = \{5, 7, 56, 23, 2902\}$$
$$t = 2963$$

Yes! $S' = \{5, 56, 2902\}$

Other Problems
SUBSET-SUM

input: A set S of integers together with an integer t

question: Is there a subset S' of S such that

$$\sum_{x \in S'} x = t?$$

Example:

$$S = \{5, 7, 56, 23, 2902\}$$
$$t = 2963$$

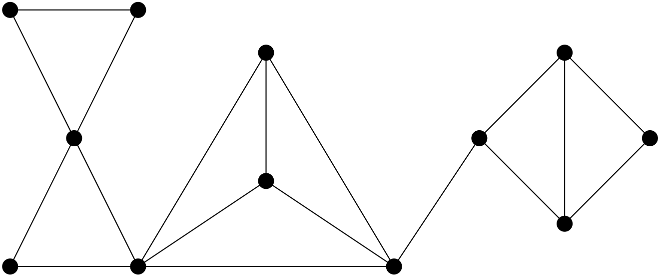
Yes! $S' = \{5, 56, 2902\}$

Not known if this problem is in P!

Other Problems
CLIQUE

input: An undirected graph $G = (V, E)$ together with an integer K

question: Does G contain a clique (complete subgraph) with K vertices?

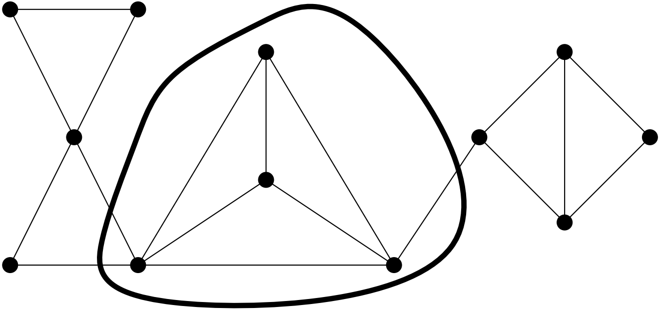


$\boxed{K = 4}$

Other Problems
CLIQUE

input: An undirected graph $G = (V, E)$ together with an integer K

question: Does G contain a clique (complete subgraph) with K vertices?



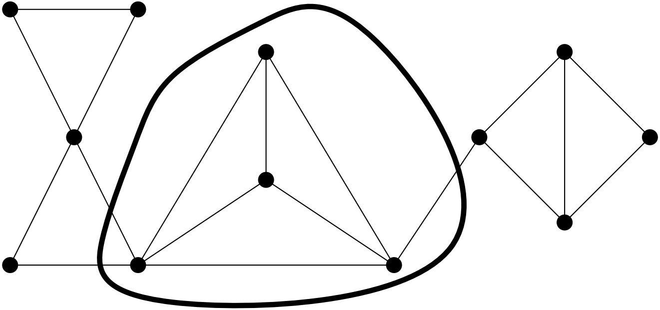
$\boxed{K = 4}$

Other Problems

CLIQUE

input: An undirected graph $G = (V, E)$ together with an integer K

question: Does G contain a clique (complete subgraph) with K vertices?



$\boxed{K = 4}$

Not known if this problem is in P!

Other Problems

For each of the 4 previous problems,

- Not known if it can be solved in polynomial time.

Other Problems

For each of the 4 previous problems,

- Not known if it can be solved in polynomial time.
- If the answer to the question is YES, then
  - There is a "short" proof for this.

Other Problems

For each of the 4 previous problems,

- Not known if it can be solved in polynomial time.
- If the answer to the question is YES, then
  - There is a "short" proof for this.

    Here, "short" means the length of the proof is "polynomial in the length of the input".

Other Problems

For each of the 4 previous problems,

- Not known if it can be solved in polynomial time.
- If the answer to the question is YES, then
  - There is a “short” proof for this.

    Here, “short” means the length of the proof is “polynomial in the length of the input”.

  - If someone gives us such a short proof, then we can “easily” verify this proof.

Other Problems

For each of the 4 previous problems,

- Not known if it can be solved in polynomial time.
- If the answer to the question is YES, then
  - There is a "short" proof for this.

    Here, "short" means the length of the proof is "polynomial in the length of the input".

  - If someone gives us such a short proof, then we can "easily" verify this proof.

    Here, "easily" means "in polynomial time".

HAM-CYCLE

proof/solution: sequence $v_1, \ldots, v_k$ of vertices.

HAM-CYCLE

proof/solution: sequence $v_1, \dots, v_k$ of vertices.

verification:

- $k = n$?

HAM-CYCLE

proof/solution: sequence $v_1, \ldots, v_k$ of vertices.

verification:

- $k = n$?
- no duplicates in $v_1, \ldots, v_k$?

HAM-CYCLE

proof/solution: sequence $v_1, \dots, v_k$ of vertices.

verification:

- $k = n$?
- no duplicates in $v_1, \dots, v_k$?
- For each $1 \leq i < k - 1$, $\{v_i, v_{i+1}\}$ is an edge?

HAM-CYCLE

proof/solution: sequence $v_1, ..., v_k$ of vertices.

verification:

- $k = n$?
- no duplicates in $v_1, ..., v_k$?
- For each $1 \leq i < k - 1$, $\{v_i, v_{i+1}\}$ is an edge?
- $\{v_k, v_1\}$ is an edge?

TSP

proof/solution: sequence $v_1, \ldots, v_k$ of vertices.

TSP

proof/solution: sequence $v_1, \ldots, v_k$ of vertices.

verification:
- $k = n$?

TSP

proof/solution: sequence $v_1, ..., v_k$ of vertices.

verification:
- $k = n$?
- no duplicates in $v_1, ..., v_k$?

TSP

proof/solution: sequence $v_1, \ldots, v_k$ of vertices.

verification:

- $k = n$?
- no duplicates in $v_1, \ldots, v_k$?
- Do we have

$$\sum_{i=1}^{k-1} wt(v_i, v_{i+1}) + wt(v_k, v_1) \leq K?$$

SUBSET-SUM

proof/solution: A set S'.

SUBSET-SUM

proof/solution: A set S'.

verification:
- Is S' a subset of S?

SUBSET-SUM

proof/solution: A set S'.

verification:
- Is S' a subset of S?
- Do we have

$$\sum_{x \in S'} x = t?$$

CLIQUE

proof/solution: A set V'.

CLIQUE

proof/solution: A set V'.

verification:
- Is V' a subset of V?

CLIQUE

proof/solution: A set V'.

verification:

- Is V' a subset of V?
- Do we have $|V'| = K$?

CLIQUE

proof/solution: A set V'.

verification:

- Is V' a subset of V?
- Do we have $|V'| = K$?
- For each $u, v \in V'$ such that $u \neq v$, is $\{u, v\}$ an edge in E?

For each of the 4 previous problems,

- Not known if it can be solved in polynomial time.

- If the answer to the question is YES, then
  - There is a "short" proof for this.

    Here, "short" means the length of the proof is "polynomial in the length of the input".

  - If someone gives us such a short proof, then we can "easily" verify this proof.

    Here, "easily" means "in polynomial time".

Complexity Class *NP*

A decision problem A is in NP if

- If for a given input I, the answer to the question $A(I)$ is YES, then there exists a proof/solution/certificate C such that
    - C is short (polynomial size in the length of I)
    - In polynomial time, we can verify that C is a correct proof for the fact that $A(I) = YES$.

Complexity Class NP

A decision problem A is in NP if

- If for a given input I, the answer to the question $A(I)$ is YES, then there exists a proof/solution/certificate C such that
  - C is short (polynomial size in the length of I)
  - In polynomial time, we can verify that C is a correct proof for the fact that $A(I) = YES$.

NP stands for *Nondeterministic Polynomial*.

Complexity Class NP

A decision problem A is in NP if

- If for a given input I, the answer to the question $A(I)$ is YES, then there exists a proof/solution/certificate C such that
    - C is short (polynomial size in the length of I)
    - In polynomial time, we can verify that C is a correct proof for the fact that $A(I) = YES$.

NP stands for *Nondeterministic Polynomial*.

The following problems are in NP:

HAM-CYCLE, TSP, SUBSET-SUM, CLIQUE