CSE211 – Formal Languages and Automata Theory

U4L13_Tractable and Intractable Problems

Dr. P. Saravanan
School of Computing
SASTRA Deemed to be University

Polynomial time algorithms: $DTIME(n^k)$

Represent tractable algorithms:

For small k we can compute the result fast

Exponential time algorithms: $DTIME(2^n)$

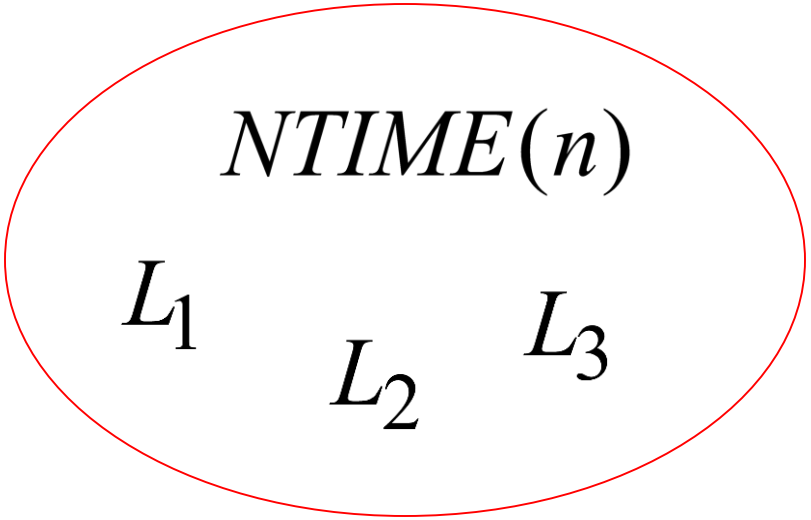
Represent intractable algorithms:

Some problem instances

may take centuries to solve

Non-Determinism

Language class: $NTIME(n)$



A Non-Deterministic Turing Machine accepts each string of length n in time $O(n)$

Example: $L = \{ww\}$

Non-Deterministic Algorithm
to accept a string ww :

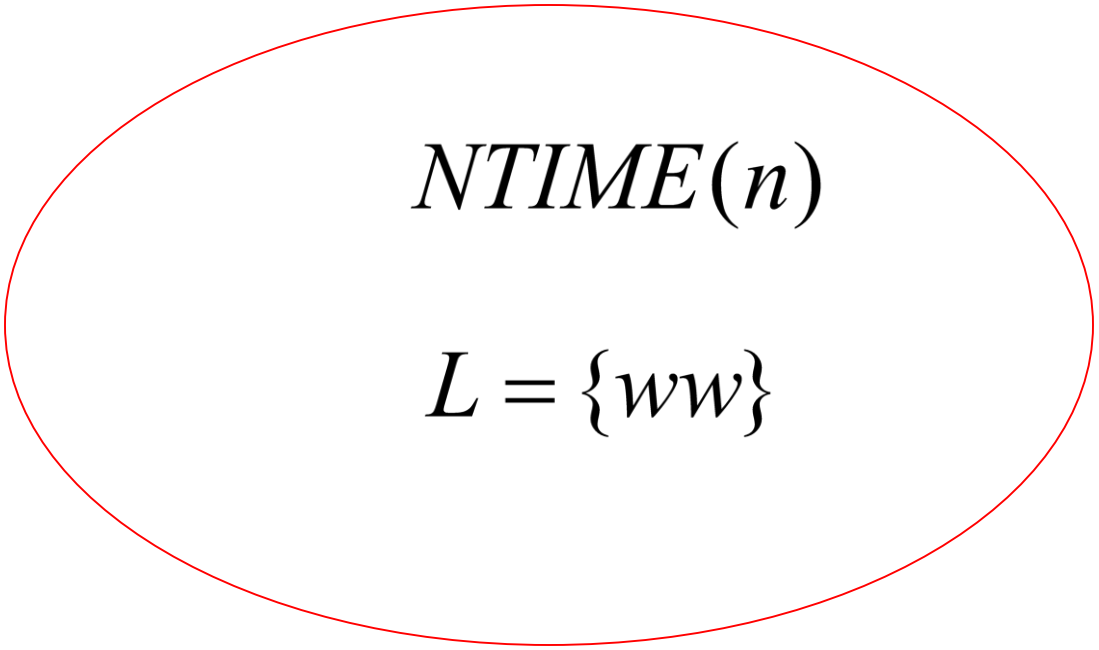
- Use a two-tape Turing machine
- Guess the middle of the string and copy w on the second tape
- Compare the two tapes

$$L = \{ww\}$$

Time needed:

- Use a two-tape Turing machine
- Guess the middle of the string and copy w on the second tape $O(|w|)$
- Compare the two tapes $O(|w|)$

Total time: $O(|w|)$


$NTIME(n)$
$L = \{ww\}$

In a similar way we define the class

$$NTIME(T(n))$$

for any time function: $T(n)$

Examples: $NTIME(n^2), NTIME(n^3),...$

Non-Deterministic Polynomial time algorithms:

$$L \in NTIME(n^k)$$

Classifications

- Algorithms that are $O(n^k)$ for some fixed k are polynomial-time algorithms.
- $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$
- reasonable, tractable

- All other algorithms are super-polynomial-time algorithms. $O(2^n)$, $O(n!)$, $O(n^n)$
- ounreasonable, intractable

Tractable and Intractable

- Tractable Problem: a problem that is solvable by a polynomial-time algorithm. The upper bound is polynomial.

- Intractable Problem: a problem that cannot be solved by a polynomial-time algorithm. The lower bound is exponential

Examples of tractable problems

(ones with known polynomial-time algorithms)

- Searching an unordered list
- Searching an ordered list
- Sorting a list
- Multiplication of integers (even though there's a gap)
- Finding a minimum spanning tree in a graph (even though there's a gap)

Intractable problems

(ones that have been proven to have no polynomial-time algorithm)

- Towers of Hanoi: we can prove that any algorithm that solves this problem must have a worst-case running time that is at least $2^n - 1$
- List all permutations (all possible orderings) of n numbers.

P class and NP class

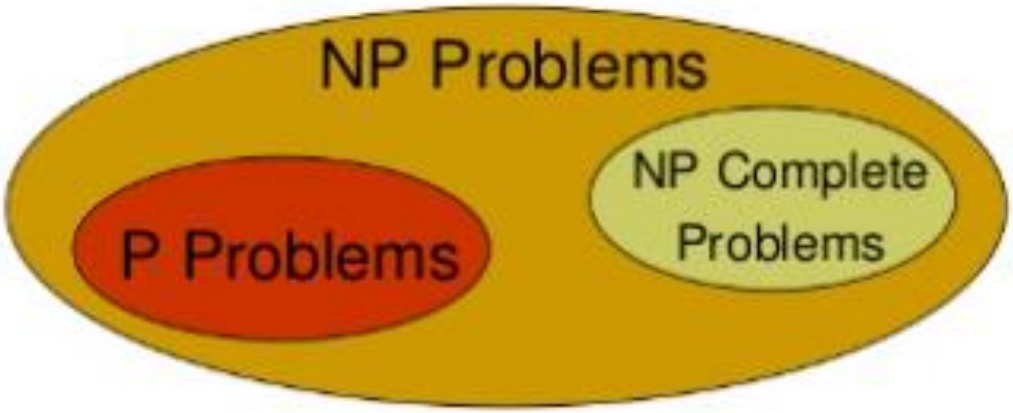
- The class P consists of all those decision problems that can be solved on a deterministic sequential machine (e.g. a computer) in an amount of time that is polynomial with respect to the size of the input

- The class NP consists of all those decision problems whose positive solutions can be verified in polynomial time given the right information

NP-Complete

- The class NPC consists of all those problems in NP that are least likely to be in P
  - Each of these problems is called NP Complete
  - Monkey puzzle, Traveling salesperson, Hamiltonian path, map coloring, satisfiability are all in NPC.
- Every problem in NPC can be transformed to another problem in NPC
  - If there were some way to solve one of these problems in polynomial time, we should be able to solve all of these problems in polynomial time.

Complexity Classes

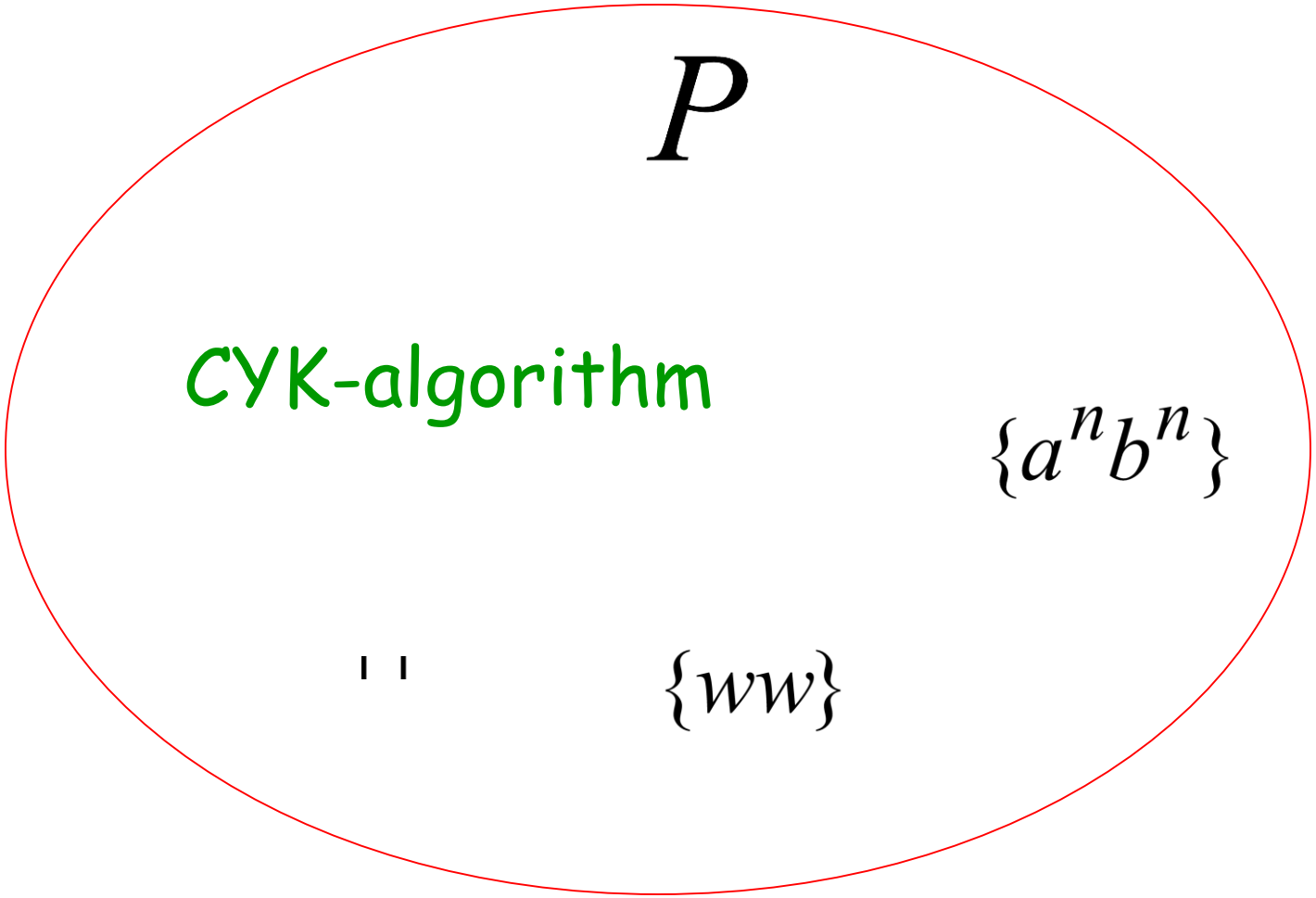


We know that P < NP, since any problem that can be solved in polynomial time can certainly have a solution verified in polynomial time.

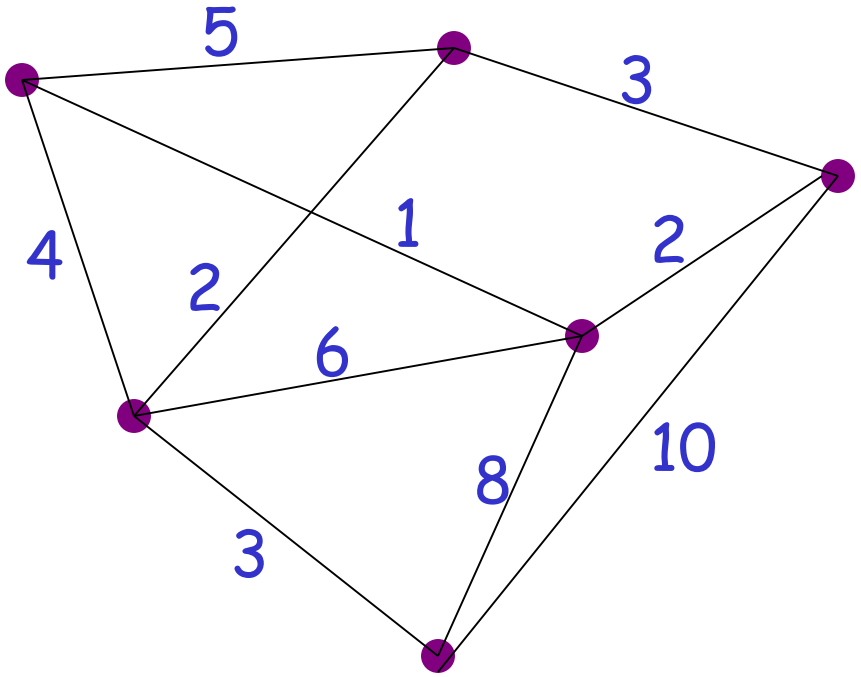
The class P

$$P = \cup DTIME(n^k) \qquad \text{for all } k$$

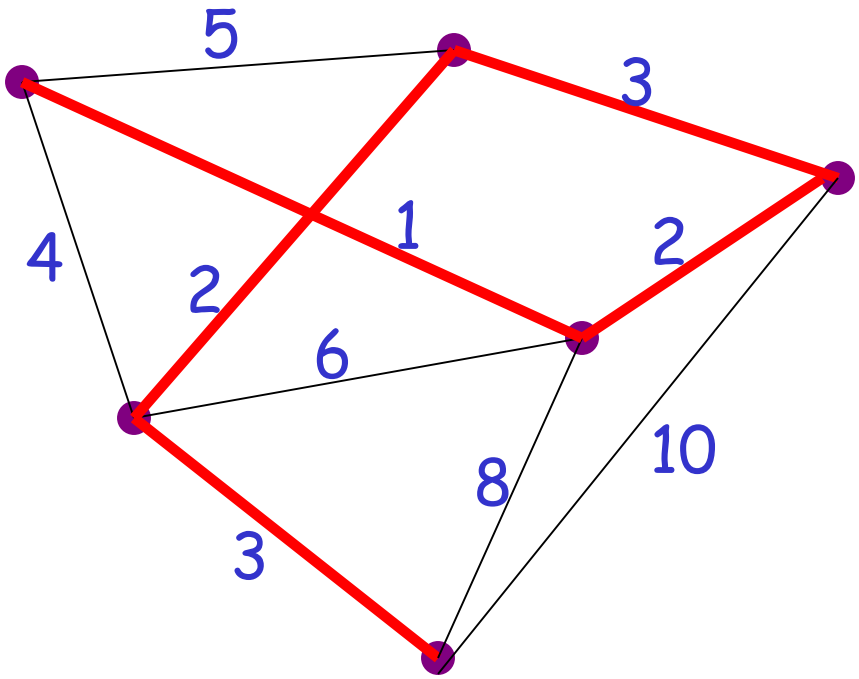
- Polynomial time
- All tractable problems


P
CYK-algorithm
$\{a^n b^n\}$
$\{ww\}$

Example: the Traveling Salesperson Problem



Question: what is the shortest route that connects all cities?



Question: what is the shortest route that connects all cities?

A solution: search exhuastively all hamiltonian paths

L = {shortest hamiltonian paths}

$$L \in DTIME(n!) \approx DTIME(2^n)$$

Exponential time

Intractable problem

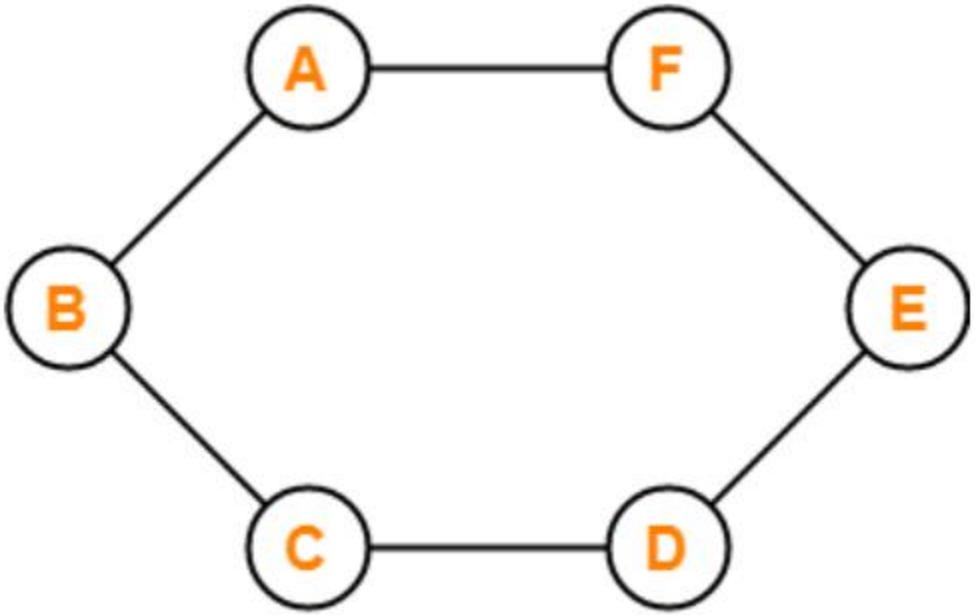
hamiltonian paths

- If there exists a closed walk in the connected graph that visits every vertex of the graph exactly once
- (except starting vertex) without repeating the edges,
- then such a graph is called as a Hamiltonian graph.

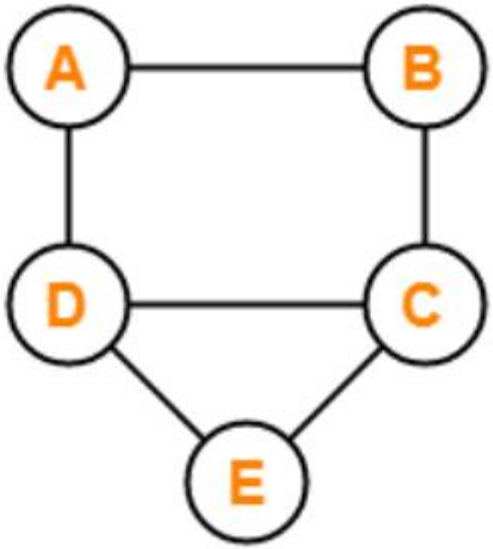
OR

- Any connected graph that contains a Hamiltonian circuit is called as a Hamiltonian Graph.

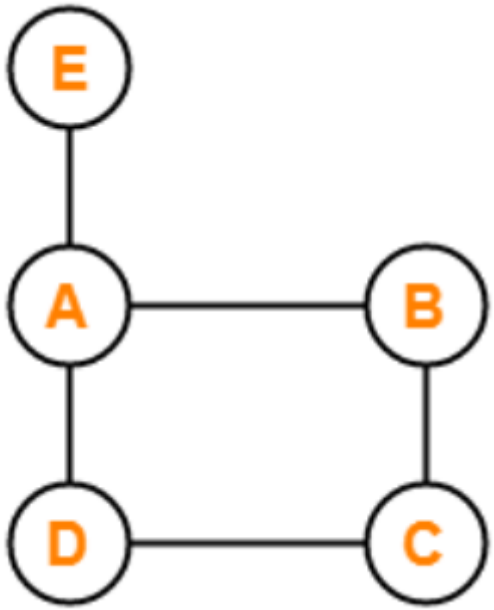
Example



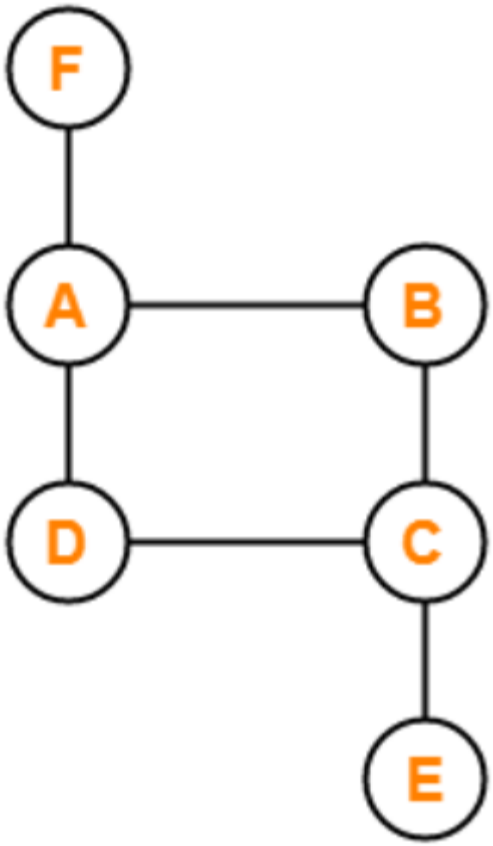
Example of Hamiltonian Graph



Hamiltonian Path = ABCDE



Hamiltonian Path = EABCD



No Hamiltanion Path Exist