CSC 339 – Theory of Computation
Fall 2023

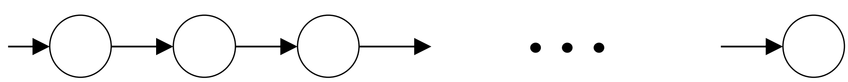
14. Time and Space Complexities

Outline

- Time complexity
- Polynomial time algorithms
- Exponential time algorithms
- Non-Determinism
- Space complexity

Consider a deterministic Turing Machine M which decides a language L.

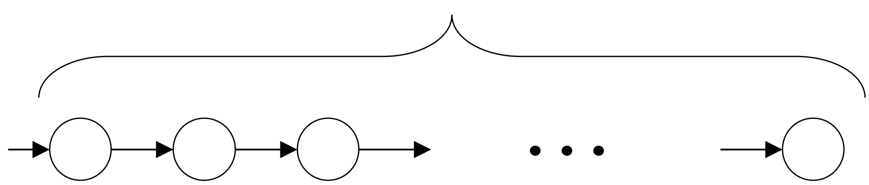
For any string w the computation of M terminates in a finite amount of transitions.



Initial state

Accept or Reject w

Decision **Time** = #transitions



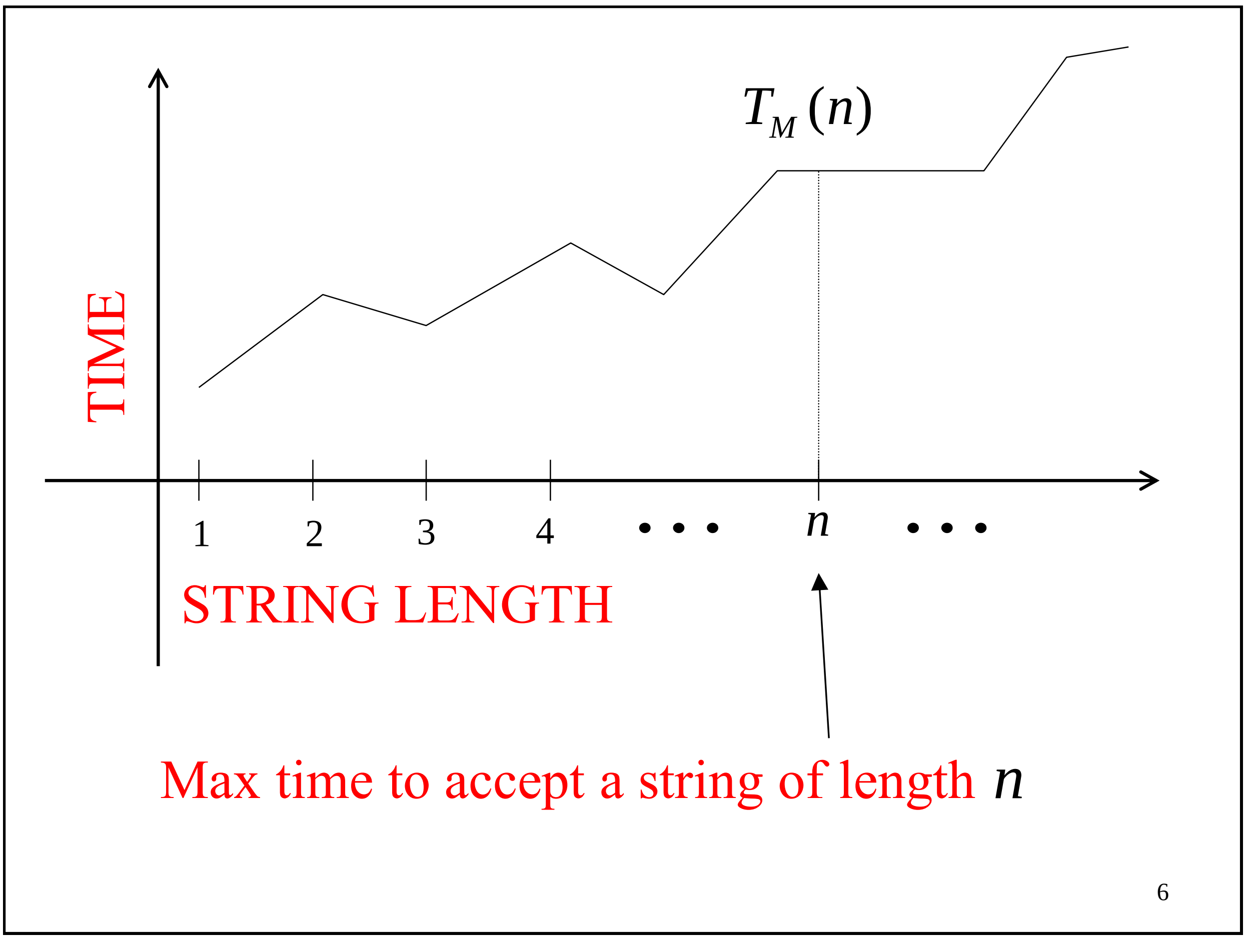
Initial state

Accept or Reject w

Consider now all strings of length n

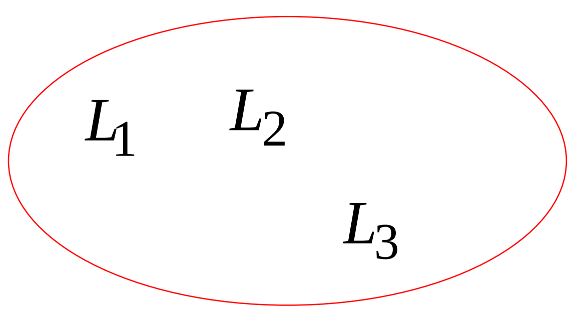
$T_M(n)$ = maximum time required to decide any string of length n

5


$T_M(n)$
TIME
1
2
3
4
• • •
n
• • •
STRING LENGTH
Max time to accept a string of length n
6

Time Complexity Class: $TIME(T(n))$

All Languages decidable by a deterministic Turing Machine in time $O(T(n))$.



Example: $L_1 = \{a^n b: \ n \geq 0\}$

This can be decided in $O(n)$ time.

$TIME(n)$

$L_1 = \{a^n b: \ n \geq 0\}$

Other example problems in the same class

$$TIME(n)$$

$$L_1 = \{a^n b : \; n \geq 0\}$$

$$\{ab^n aba : \; n, k \geq 0\}$$

$$\{b^n : \; n \text{ is even}\}$$

$$\{b^n : \; n = 3k\}$$

9

Examples in class:

$$TIME(n^2)$$

$$\{a^n b^n : n \geq 0\}$$

$$\{ww^R : \ w \in \{a,b\}\}$$

$$\{ww : \ w \in \{a,b\}\}$$

10

Examples in class:

$$TIME(n^3)$$

$$L_2 = \{\langle G, w\rangle : w \text{ is generated by context-free grammar } G\}$$

$$L_3 = \{\langle M_1, M_2, M_3\rangle : n \times n \text{ matrices and } M_1 \times M_2 = M_3\}$$

11

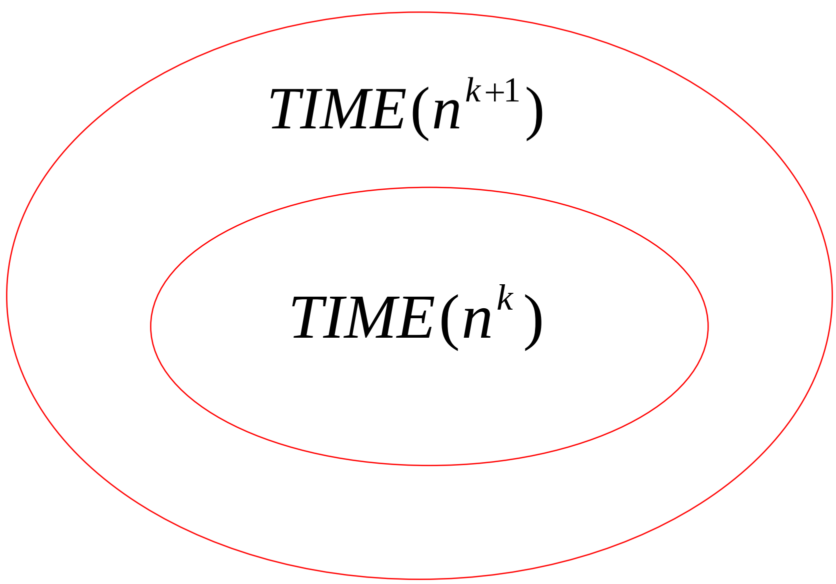
Polynomial time algorithms: $TIME(n^k)$

With constant $k > 0$

Represents tractable algorithms:

For small k we can decide the result fast.

It can be shown: $TIME(n^k) \subset TIME(n^{k+1})$



The Time Complexity Class P

$$P = \bigcup_{k>0} TIME(n^k)$$

Represents:

- Polynomial time algorithms.
- "Tractable" problems.

14

Class P

$\{a^n b\}$

$\{a^n b^n\}$

$\{ww\}$

Matrix multiplication

…

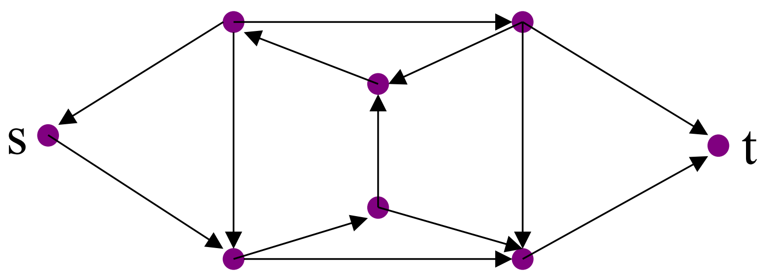
Exponential time algorithms: $TIME(2^{n^k})$

Represent intractable algorithms:

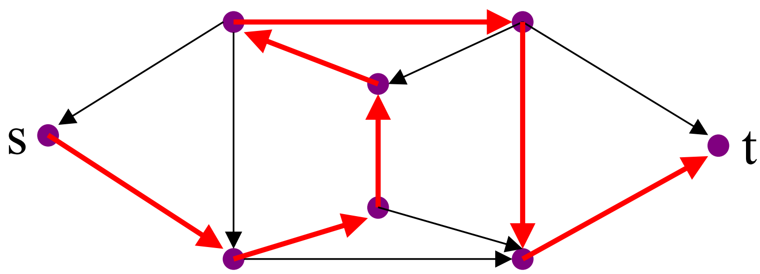
Some problem instances may take centuries to solve.

16

Example 1: The Hamiltonian Path Problem



Question: is there a Hamiltonian path from s to t?



YES!

18

A solution: search exhaustively all paths

L = {<G,s,t>: there is a Hamiltonian path in G from s to t}

$$L \in TIME(n!) \approx TIME(2^{n^k})$$

Exponential time

Intractable problem

Example 2: The Satisfiability Problem

Boolean expressions in
Conjunctive Normal Form (CNF):

$$t_1 \wedge t_2 \wedge t_3 \wedge \cdots \wedge t_k$$ Clauses

$$t_i = x_1 \vee \bar{x}_2 \vee x_3 \vee \cdots \vee \bar{x}_p$$ Variables

Question: is the expression satisfiable?

Example: $(\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3)$

Satisfiable: $x_1 = 0, \; x_2 = 1, \; x_3 = 1$

$(\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_3) = 1$

Example: $(x_1 \vee x_2) \wedge \bar{x}_1 \wedge \bar{x}_2$

Not satisfiable

22

$L = \{w : \text{expression } w \text{ is satisfiable}\}$

$L \in TIME(2^{n^k})$ Exponential

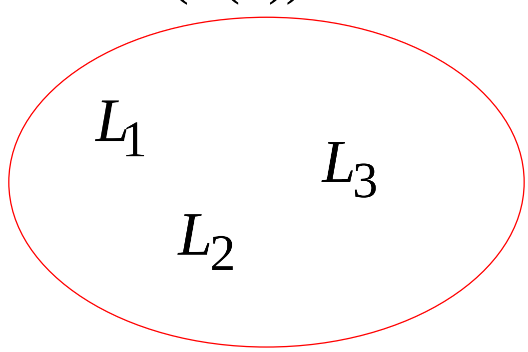
Algorithm:

Search exhaustively all the possible binary values of the variables.

23

Non-Determinism

Language class: $NTIME(T(n))$



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

Non-Deterministic Polynomial time algorithms:

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

The class NP

Non-Deterministic Polynomial time

$$NP = \bigcup_{k>0} NTIME(n^k)$$

26

Example: The satisfiability problem

$$L = \{w : \text{expression } w \text{ is satisfiable}\}$$

Non-Deterministic algorithm:

- Guess an assignment of the variables.
- Check if this is a satisfying assignment.

$$L = \{w : \text{expression } w \text{ is satisfiable}\}$$

Time for n variables:

- Guess an assignment of the variables: $O(n)$
- Check if this is a satisfying assignment: $O(n)$

Total time: $O(n)$

$L = \{w : \text{expression } w \text{ is satisfiable}\}$

$L \in NP$

The satisfiability problem is an *NP*- Problem.

29

Observation:

$$P \subseteq NP$$

Deterministic Polynomial

Non-Deterministic Polynomial

Open Problem: $P = NP$?

Example: Does the Satisfiability problem have a polynomial time deterministic algorithm?

We don't know the answer, so far.

Space complexity

- Let M be a deterministic Turing machine that halts on all inputs.
- The space complexity of M is the function:

  $f\colon \mathbb{N} \rightarrow \mathbb{N}$, where $f(n)$ is the maximum number of tape cells that M scans on any input of length n.
- We say that M runs in space $f(n)$.

32

Space complexity

- If M is a nondeterministic Turing machine wherein all branches halt on all inputs.
- The space complexity of M is the function:

  $f : \mathbb{N} \rightarrow \mathbb{N}$, where $f(n)$ is the maximum number of tape cells that M scans on any branch of its computation for any input of length n.

Space complexity class $\mathrm{SPACE}\big(f(n)\big)$

- Let $f\colon\ \mathbb{N} \to \mathbb{R}^{+}$ be a function.
- The space complexity class $\mathrm{SPACE}\big(f(n)\big)$ is defined by:

$\mathrm{SPACE}\big(f(n)\big) = \{L \mid L$ is a language decided by an $O\big(f(n)\big)$ space deterministic Turing machine$\}$

34

Space complexity class $\mathrm{NSPACE}\big(f(n)\big)$

- Let $f\colon \mathbb{N} \to \mathbb{R}^+$ be a function.
- The space complexity class $\mathrm{NSPACE}\big(f(n)\big)$ is defined by:

$\mathrm{NSPACE}\big(f(n)\big) = \{L | L$ is a language decided by an $O\big(f(n)\big)$ space nondeterministic Turing machine$\}$

35

Savitch's theorem

- For any function $f\colon \mathbb{N} \to \mathbb{R}^+$, where $f(n) \geq n$,

  $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$

Space complexity class PSPACE

- PSPACE is the class of languages that are decidable in polynomial space on a deterministic Turing machine:

$$\mathrm{PSPACE} = \bigcup_{k} \mathrm{SPACE}(n^k)$$