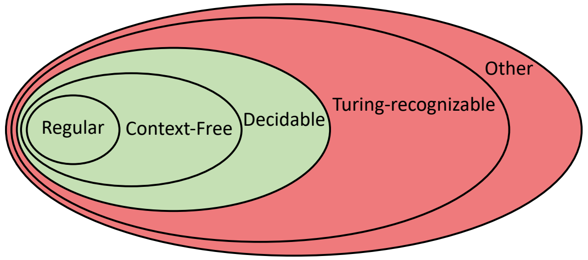
# P and NP
## CSCI 338

# Announcements

- Test 2. (AVG = 72%, MED = 85%)
- Project 2. (AVG = 80%, MED = 100%)

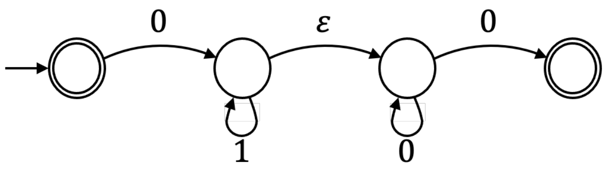
Computability: What’s solvable by computers.



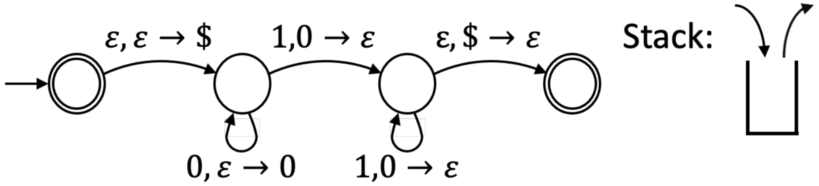

**January** **May**

Computational Models

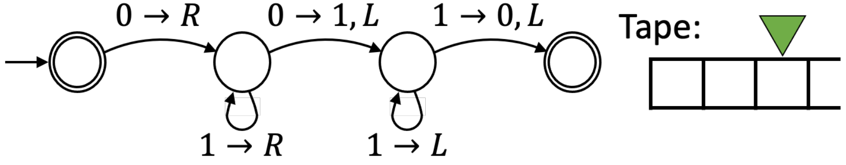
NFA




PDA




Turing Machine




Complexity: What’s efficiently solvable by computers.

Goal: Understand and identify fundamental limitations of computers.

# Computational Complexity

Determining the amount of resources required to accomplish some task (solve a problem).

- Time (most common)
- Space (close behind)
- Power (sensor networks, spacecraft, military)
- Network (Netflix, IoT)

# Definitions

Definition: For TM $M$, the **running time of $\boldsymbol{M}$** is the function $f: \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum number of steps $M$ used on any input of length $n$.

# Definitions

Definition: For TM $M$, the **running time of $\boldsymbol{M}$** is the function $f\colon \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum number of steps $M$ used on any input of length $n$.

Definition: Let $f$ and $g$ be functions $f, g\colon \mathbb{N} \to \mathbb{R}^+$, $\boldsymbol{f(n) \in O\big(g(n)\big)}$ if $\exists$ positive integers $c$ and $n_0$ such that $\forall\, n \geq n_0$, $f(n) \leq c\, g(n)$.

Asymptotic upper bound on running time.

# P

Definition: Let $t: \mathbb{N} \rightarrow \mathbb{R}^{+}$ be a function. The time complexity class, **TIME($\boldsymbol{t(n)}$)**, is the collection of all languages that are decidable by an $O(t(n))$ time TM.

Definition: $\boldsymbol{P}$ is the set of languages that are decidable in polynomial time on a deterministic single-tape TM.

$$P = \bigcup_{k} \text{TIME}\left(n^{k}\right)$$

Properties of $P$:

- Inclusion in $P$ holds for all computational models polynomially equivalent to deterministic, single-tape TMs.
- Roughly corresponds to problems solvable by a computer.

# P

Definition: Let $t: \mathbb{N} \rightarrow \mathbb{R}^+$ be a function. The time complexity class, **TIME($\boldsymbol{t(n)}$)**, is the collection of all languages that are decidable by an $O(t(n))$ time TM.

Definition: $\boldsymbol{P}$ is the set of languages that are decidable in polynomial time on a deterministic single-tape TM.

$$P = \bigcup_k \text{TIME}(n^k)$$

Properties of $P$:

- Inc **How do you show something is in $\boldsymbol{P}$?**
  po e TMs.
- Roughly corresponds to problems solvable by a computer.

# P

Definition: Let $t: \mathbb{N} \to \mathbb{R}^+$ be a function. The time complexity class, **TIME($\boldsymbol{t(n)}$)**, is the collection of all languages that are decidable by an $O(t(n))$ time TM.

Definition: $\boldsymbol{P}$ is the set of languages that are decidable in polynomial time on a deterministic single-tape TM.

$$P = \bigcup_k \text{TIME}(n^k)$$

Properties of $P$:

- Inc… po… e TMs.
- Roughly corresponds to problems solvable by a computer.

**How do you show something is in $\boldsymbol{P}$?**

**Build a polynomial time decider for it.**

# Why Polynomial?

Given a problem of size $n$...

Algorithm A solves it in $n^2$ seconds

Algorithm B solves it in $2^n$ seconds

# Why Polynomial?

Given a problem of size $n$…

Algorithm A solves it in $n^2$ seconds

Algorithm B solves it in $2^n$ seconds

Would you rather have a minute of compute time for algorithm A, or five minutes of compute time for algorithm B?

# Why Polynomial?

Given a problem of size $n$…

| Algorithm A solves it in $n^2$ seconds | Algorithm B solves it in $2^n$ seconds |
|---|---|

Would you rather have a minute of compute time for algorithm A, or five minutes of compute time for algorithm B?

A solves in $1$ min $\Rightarrow$ A solves in $60$ sec $\Rightarrow n \approx 7.7$

# Why Polynomial?

Given a problem of size $n$...

Algorithm A solves it in $n^2$ seconds

Algorithm B solves it in $2^n$ seconds

Would you rather have a minute of compute time for algorithm A, or five minutes of compute time for algorithm B?

A solves in 1 min $\Rightarrow$ A solves in 60 sec $\Rightarrow n \approx 7.7$
B solves in 5 min $\Rightarrow$ B solves in 300 sec $\Rightarrow n \approx 8.2$

# Why Polynomial?

Given a problem of size $n$…

Algorithm A solves it in $n^2$ seconds

Algorithm B solves it in $2^n$ seconds

Would you rather have an hour of compute time for algorithm A, or 10 billion years of compute time for algorithm B?

# Why Polynomial?

Given a problem of size $n$…

| Algorithm A solves it in $n^2$ seconds | Algorithm B solves it in $2^n$ seconds |
|---|---|

Would you rather have an hour of compute time for algorithm A, or 10 billion years of compute time for algorithm B?

A solves in 1 hr $\Rightarrow$ A solves in $3600$ sec $\Rightarrow n = 60$

# Why Polynomial?

Given a problem of size $n$…

Algorithm A solves it in $n^2$ seconds

Algorithm B solves it in $2^n$ seconds

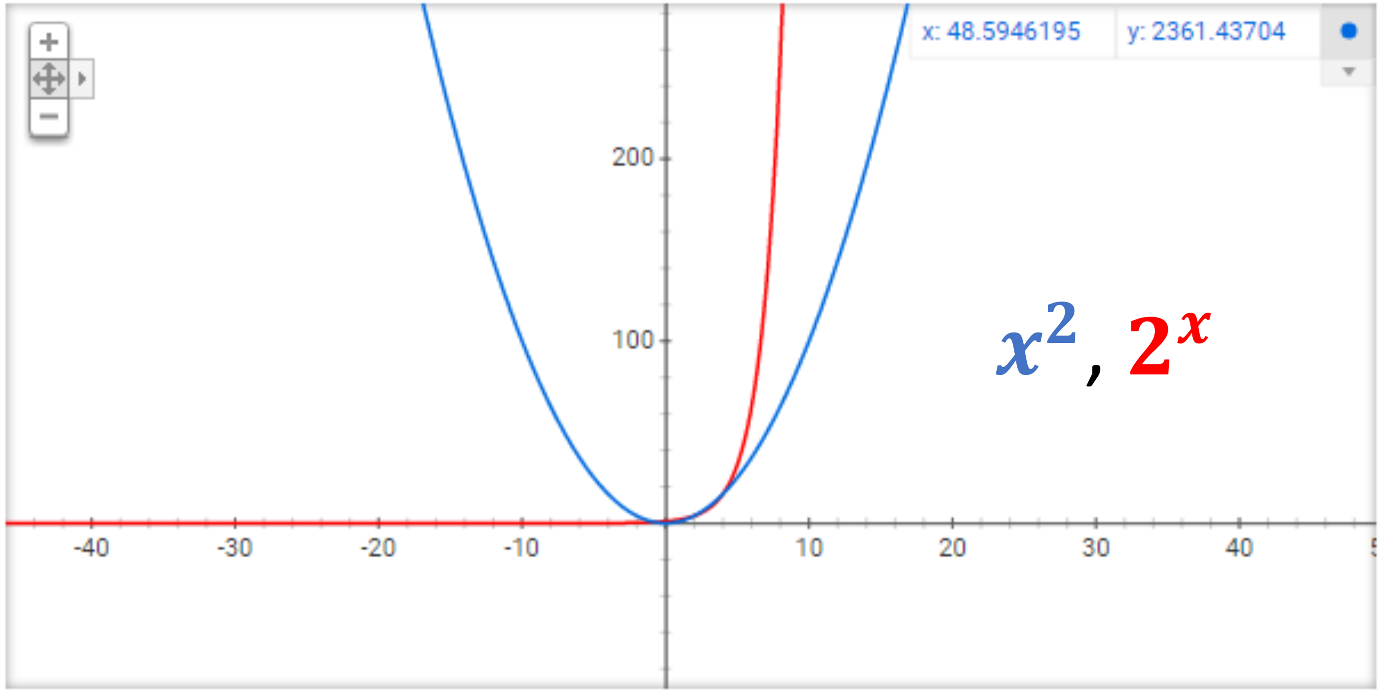
Would you rather have an hour of compute time for algorithm A, or 10 billion years of compute time for algorithm B?

A solves in $1$ hr $\Rightarrow$ A solves in $3600$ sec $\Rightarrow n = 60$

B solves in $10\text{B}$ yr $\Rightarrow$ B solves in lots of sec $\Rightarrow n \approx 58$

315360000000000000

# Why Polynomial?




$x^2$, $2^x$

Polynomial time algorithms are usually useful,
exponential time algorithms are rarely useful

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof:

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.

$N$ = on input $\langle G, s, t\rangle$

1.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.

$N$ = on input $\langle G, s, t\rangle$

1. Mark $s$.
2. 

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.

$N$ = on input $\langle G, s, t\rangle$

1. Mark $s$.
2. Repeat until no new nodes are marked:
3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.
4.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.

$N$ = on input $\langle G, s, t\rangle$

1. Mark $s$.
2. Repeat until no new nodes are marked:
3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.
4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.

$N$ = on input $\langle G, s, t\rangle$

1. Mark $s$.
2. Repeat until no new nodes are marked:
3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.
4. If $t$ is marked, accept. Otherwise, reject.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

1. Mark $s$.
2. Repeat until no new nodes are marked:
3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.
4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

**?** → 1. Mark $s$.

2. Repeat until no new nodes are marked:
3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.
4. If $t$ is marked, accept. Otherwise, reject.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)}$ → 1. Mark $s$.

2. Repeat until no new nodes are marked:
3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.
4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O}(\mathbf{1})$ → 1. Mark $s$.

2. Repeat until no new nodes are marked:

? → 3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.

4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?
What graph has the most number of edges?

$\boldsymbol{O}$

.

Need to show N is a decider and runs in polynomial time.

# $PATH$

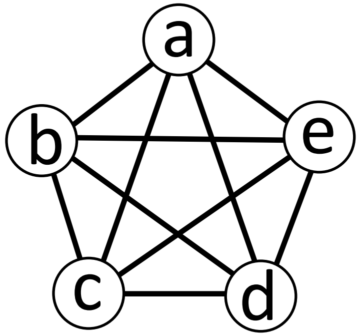
Claim: $PATH = \{\langle G, s, t \rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

Complete graph (every pair of vertices have an edge).

$\boldsymbol{O}$




.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t \rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

Complete graph (every pair of vertices have an edge).

How many edges does a complete graph with $n$ vertices have?

How many edges leave each vertex?

$\boldsymbol{O}$

.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

Complete graph (every pair of vertices have an edge).

How many edges does a complete graph with $n$ vertices have?

How many edges leave each vertex? $n - 1$

How much does that all add up to?

$\boldsymbol{O}$

.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

Complete graph (every pair of vertices have an edge).

How many edges does a complete graph with $n$ vertices have?

How many edges leave each vertex? $n - 1$

How much does that all add up to? $n(n - 1)$

Did we double count any edges?

$\boldsymbol{O}$

.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

Complete graph (every pair of vertices have an edge).

How many edges does a complete graph with $n$ vertices have?

How many edges leave each vertex? $n - 1$

How much does that all add up to? $n(n-1)$

Did we double count any edges? Yes

So how many edges are there?

$\boldsymbol{O}$

.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

Complete graph (every pair of vertices have an edge).

How many edges does a complete graph with $n$ vertices have?

How many edges leave each vertex? $n - 1$

How much does that all add up to? $n(n-1)$

Did we double count any edges? Yes

So how many edges are there? $\frac{n(n-1)}{2} \in O(n^2)$

$\boldsymbol{O}$ .

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

Complete graph (every pair of vertices have an edge).

How many edges does a complete graph with $n$ vertices have?

How many edges leave each vertex? $n - 1$

How much does that all add up to? $n(n-1)$

Did we double count any edges? Yes

So how many edges are there? $\frac{n(n-1)}{2} \in O(n^2)$

What if it is directed?

$\boldsymbol{O}$ .

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

At most, how many edges are in an undirected graph with $n$ vertices?

What graph has the most number of edges?

  Complete graph (every pair of vertices have an edge).

How many edges does a complete graph with $n$ vertices have?

  How many edges leave each vertex? $n - 1$

  How much does that all add up to? $n(n-1)$

  Did we double count any edges? Yes

  So how many edges are there? $\frac{n(n-1)}{2} \in O(n^2)$

What if it is directed? $n(n-1) \in O(n^2)$ (no double counts)

$\boldsymbol{O}$ .

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)}$ → 1. Mark $s$.

2. Repeat until no new nodes are marked:

$\boldsymbol{O(n^2)}$ → 3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.

4. If $t$ is marked, accept. Otherwise, reject.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)}$ → 1. Mark $s$.

**?** → 2. Repeat until no new nodes are marked:

$\boldsymbol{O(n^2)}$ → 3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.

4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)}$ ⟶ 1. Mark $s$.

$\boldsymbol{O(n)}$ ⟶ 2. Repeat until no new nodes are marked:

$\boldsymbol{O(n^2)}$ ⟶ 3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.

4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)}$ → 1. Mark $s$.

$\boldsymbol{O(n)}$ → 2. Repeat until no new nodes are marked:

$\boldsymbol{O(n^2)}$ → 3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.

**?** → 4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)}$ → 1. Mark $s$.

$\boldsymbol{O(n)}$ → 2. Repeat until no new nodes are marked:

$\boldsymbol{O(n^2)}$ → 3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.

$\boldsymbol{O(1)}$ → 4. If $t$ is marked, <u>accept</u>. Otherwise, <u>reject</u>.

Need to show N is a decider and runs in polynomial time.

# $PATH$

Claim: $PATH = \{\langle G, s, t\rangle : G = (V, E)$ is a directed graph with a path from $s$ to $t\} \in P$.

Proof: Build a polynomial time decider.
Suppose $|V| = n$.

$N$ = on input $\langle G, s, t\rangle$

$\boldsymbol{O(1)}$ → 1. Mark $s$.

$\boldsymbol{O(n)}$ → 2. Repeat until no new nodes are marked:

$\boldsymbol{O(n^2)}$ → 3. For each $e = (a, b) \in E$, if $a$ is marked, mark $b$.

$\boldsymbol{O(1)}$ → 4. If $t$ is marked, accept. Otherwise, reject.

N is a decider and runs in $O(n^3)$ time, therefore $PATH \in P$.