Two Big Questions: P vs. NP and P vs. BPP

Jeff Kinne

University of Wisconsin-Madison

Indiana State University, March 26, 2010

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?**

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others
- **Is randomness powerful?**

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others
- **Is randomness powerful?** P vs. BPP, L vs. BPL

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others
- **Is randomness powerful?** P vs. BPP, L vs. BPL
  - Conjecture: P=BPP, L=BPL

Computational Complexity Theory

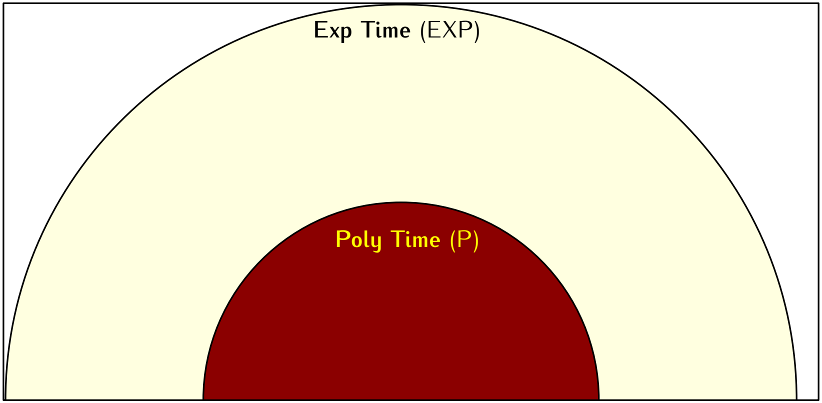
How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others
- **Is randomness powerful?** P vs. BPP, L vs. BPL
  - Conjecture: P=BPP, L=BPL
  - My work: derandomization, hierarchy theorems

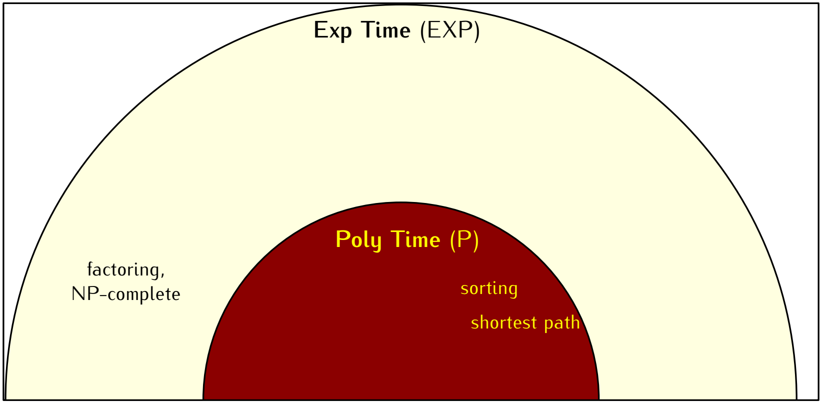
Introducing two Big Questions

Time Complexity

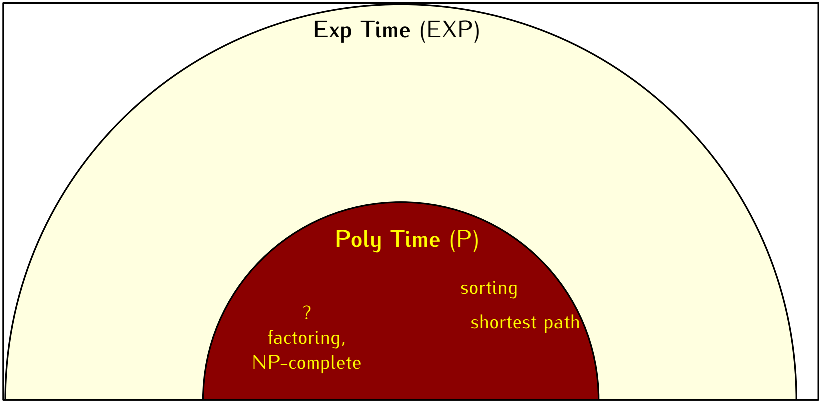
Time Complexity



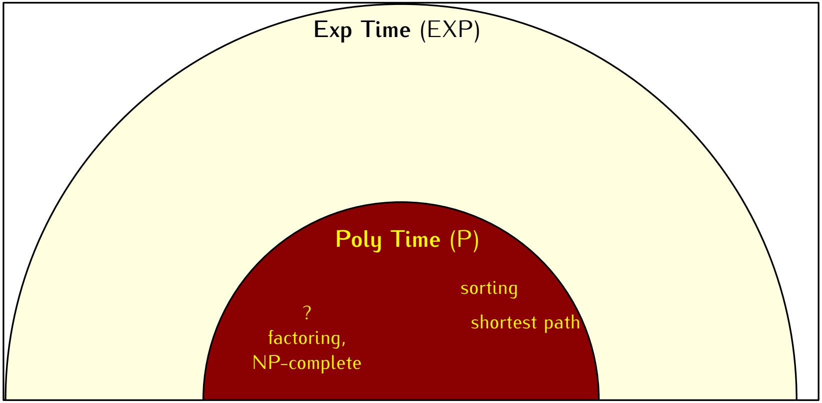
Time Complexity



Time Complexity



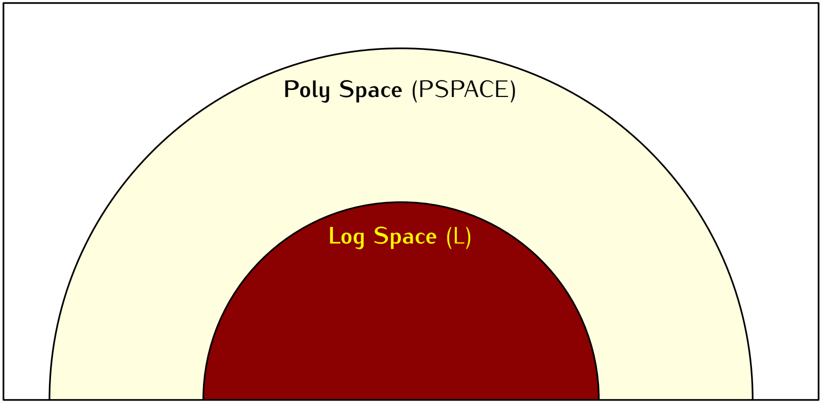
Time Complexity



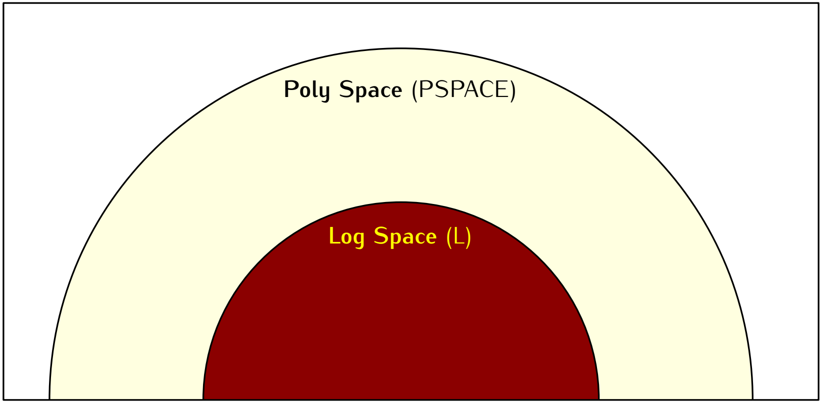
Decision Problem: yes/no questions

Memory Space Complexity

Memory Space Complexity

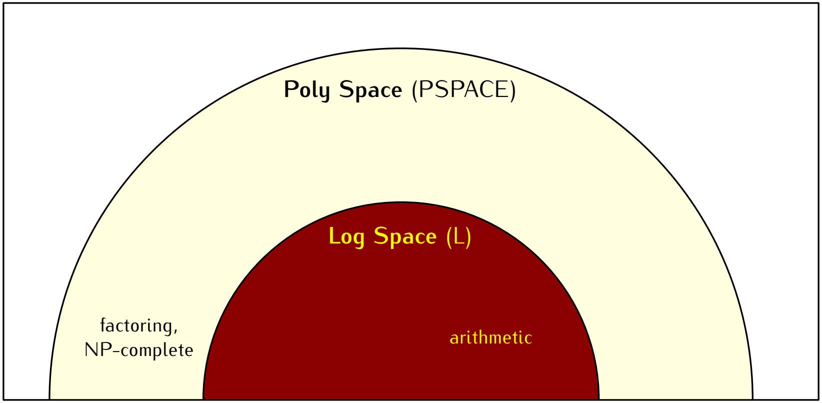


Memory Space Complexity



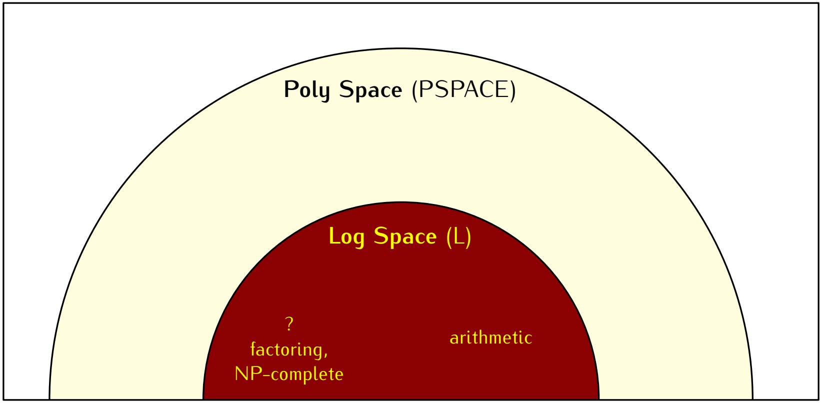
Amount of working space memory needed

Memory Space Complexity



Amount of working space memory needed

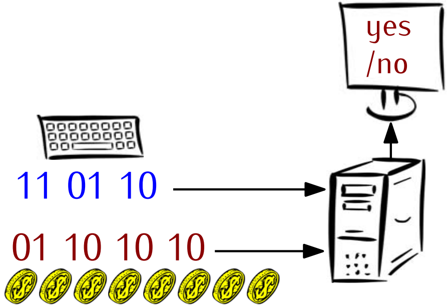
Memory Space Complexity



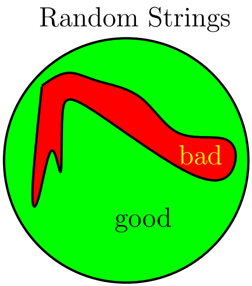
Amount of working space memory needed

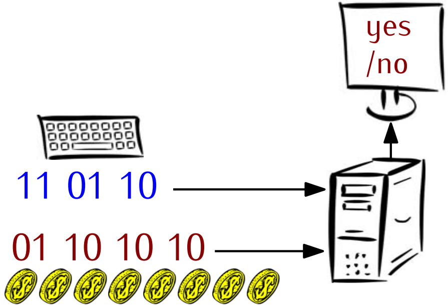
Randomized Algorithm

Randomized Algorithm

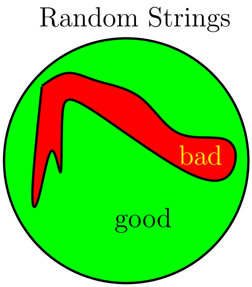


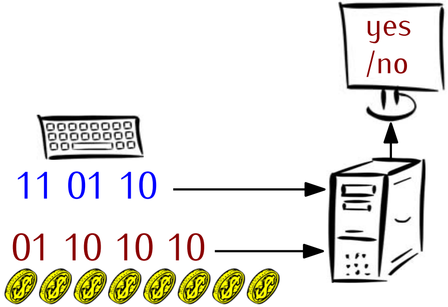
Randomized Algorithm





Randomized Algorithm





Bounded error: Correct with probability $> 99\%$

Polynomial Identity Testing

Polynomial Identity Testing

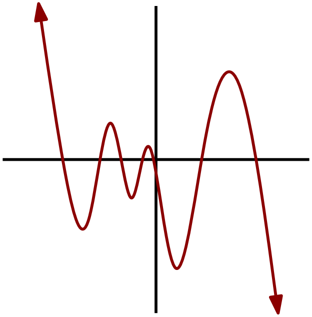
- $p(x) = x^3 \cdot (3x - x^2)^2 - x^4 \cdot (2x^3 + 5x) + x \cdot (4x^2 - x)^3$

Polynomial Identity Testing

- $p(x) = x^3 \cdot (3x - x^2)^2 - x^4 \cdot (2x^3 + 5x) + x \cdot (4x^2 - x)^3$
- **Do all terms cancel?**

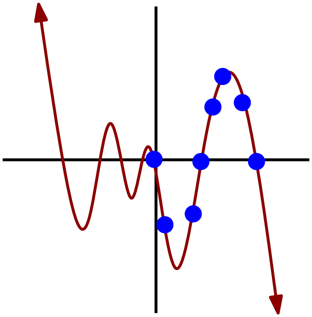
Polynomial Identity Testing

- $p(x) = x^3 \cdot (3x - x^2)^2 - x^4 \cdot (2x^3 + 5x) + x \cdot (4x^2 - x)^3$
- **Do all terms cancel?**



Polynomial Identity Testing

- $p(x) = x^3 \cdot (3x - x^2)^2 - x^4 \cdot (2x^3 + 5x) + x \cdot (4x^2 - x)^3$
- **Do all terms cancel?**



Multi-variate Polynomial Identity Testing

Multi-variate Polynomial Identity Testing

- $p(x_1, x_2, x_3, x_4) = x_4^5 \cdot (x_1 - x_2 - x_3)^{30} + (x_4 + x_3 - x_1)^{15} \cdot (x_3 - x_2 + x_1)^{20} - (x_2 - x_3 + x_4)^{20} \cdot (x_2 + x_1)^{15}$

Multi-variate Polynomial Identity Testing

- $p(x_1, x_2, x_3, x_4) = x_4^5 \cdot (x_1 - x_2 - x_3)^{30} + (x_4 + x_3 - x_1)^{15} \cdot (x_3 - x_2 + x_1)^{20} - (x_2 - x_3 + x_4)^{20} \cdot (x_2 + x_1)^{15}$
- **Do all terms cancel?**

Multi-variate Polynomial Identity Testing

- $p(x_1, x_2, x_3, x_4) = x_4^5 \cdot (x_1 - x_2 - x_3)^{30} + (x_4 + x_3 - x_1)^{15} \cdot (x_3 - x_2 + x_1)^{20} - (x_2 - x_3 + x_4)^{20} \cdot (x_2 + x_1)^{15}$
- **Do all terms cancel?**

Randomized Algorithm

Multi-variate Polynomial Identity Testing

- $p(x_1, x_2, x_3, x_4) = x_4^5 \cdot (x_1 - x_2 - x_3)^{30} + (x_4 + x_3 - x_1)^{15} \cdot (x_3 - x_2 + x_1)^{20} - (x_2 - x_3 + x_4)^{20} \cdot (x_2 + x_1)^{15}$
- **Do all terms cancel?**

Randomized Algorithm

- Pick point $(x_1, ..., x_4)$, each $x_i \in_R S$

Multi-variate Polynomial Identity Testing

- $p(x_1, x_2, x_3, x_4) = x_4^5 \cdot (x_1 - x_2 - x_3)^{30} + (x_4 + x_3 - x_1)^{15} \cdot (x_3 - x_2 + x_1)^{20} - (x_2 - x_3 + x_4)^{20} \cdot (x_2 + x_1)^{15}$
- **Do all terms cancel?**

Randomized Algorithm

- Pick point $(x_1, ..., x_4)$, each $x_i \in_R S$
- p non-zero, degree d

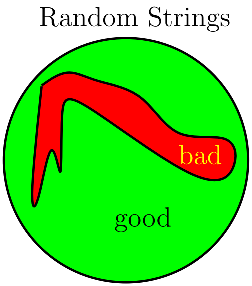
Multi-variate Polynomial Identity Testing

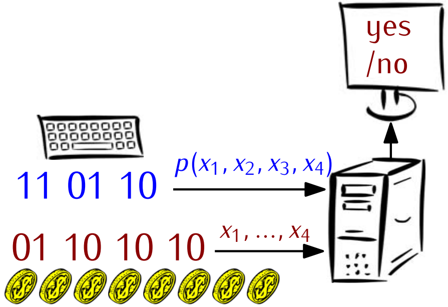
- $p(x_1, x_2, x_3, x_4) = x_4^5 \cdot (x_1 - x_2 - x_3)^{30} + (x_4 + x_3 - x_1)^{15} \cdot (x_3 - x_2 + x_1)^{20} - (x_2 - x_3 + x_4)^{20} \cdot (x_2 + x_1)^{15}$
- **Do all terms cancel?**

Randomized Algorithm

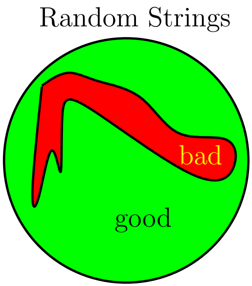
- Pick point $(x_1, ..., x_4)$, each $x_i \in_R S$
- p non-zero, degree $d \Rightarrow \Pr[p(x_1, ..., x_4) = 0] \leq \frac{d}{|S|}$

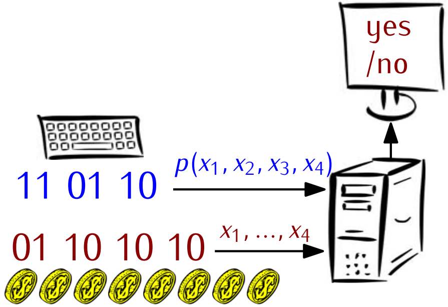
Randomized Algorithm





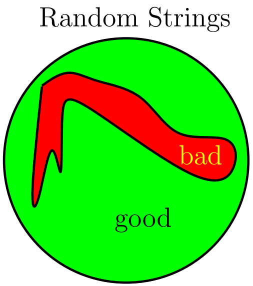
Randomized Algorithm

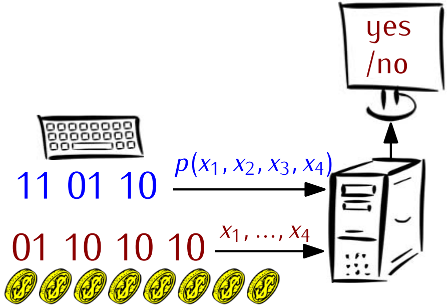




- **BPP**: Bounded-error Probabilistic Poly time

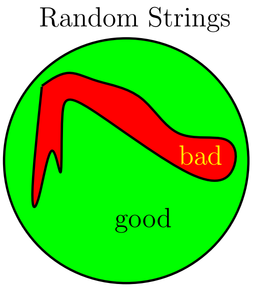
Randomized Algorithm

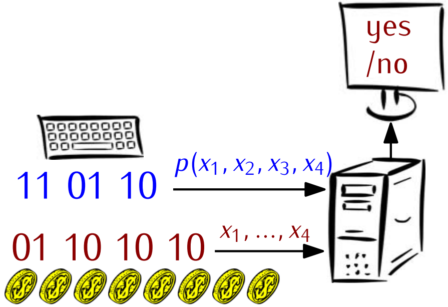




- **BPP**: Bounded-error Probabilistic Poly time , **BPL**: log space

Randomized Algorithm

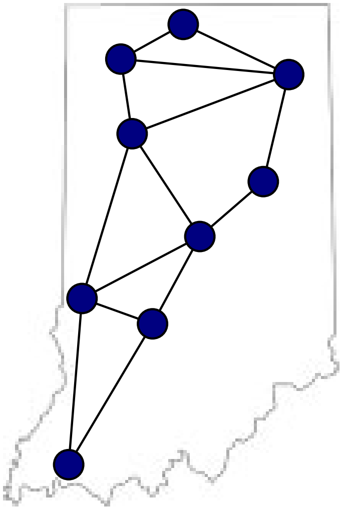




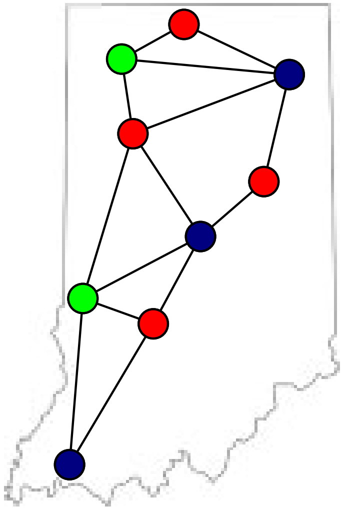
- **BPP**: Bounded-error Probabilistic Poly time , **BPL**: log space
- **P $\overset{?}{=}$ BPP**

Nondeterministic Algorithm

Graph 3-Coloring

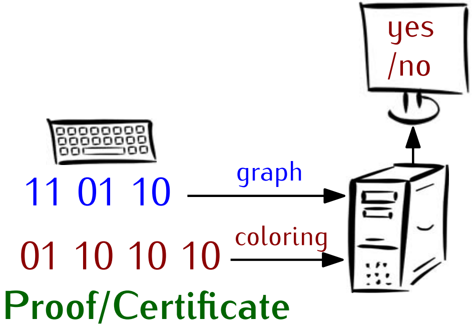


Graph 3-Coloring

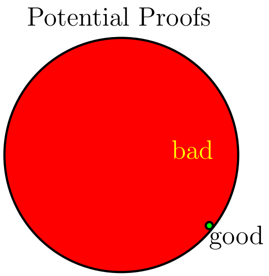


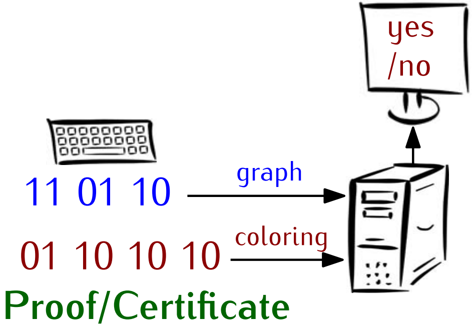
Nondeterministic Algorithm

Nondeterministic Algorithm

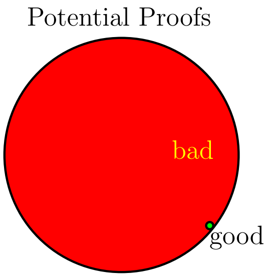


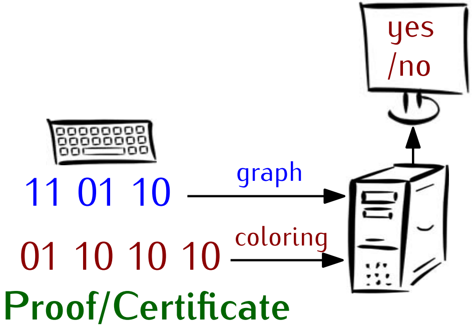
Nondeterministic Algorithm





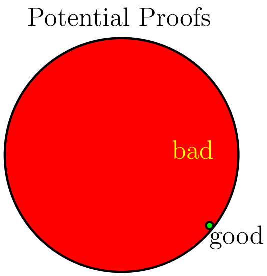
Nondeterministic Algorithm

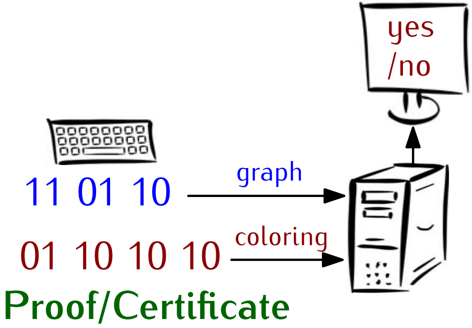




- **NP**: Nondeterministic Polynomial time

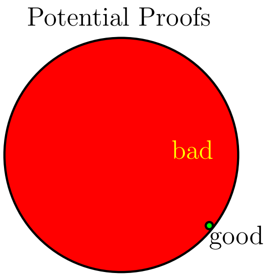
Nondeterministic Algorithm

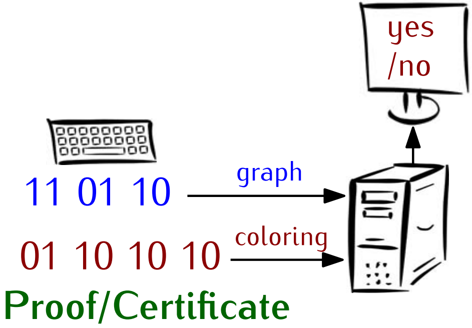




- **NP**: Nondeterministic Polynomial time
- “NP-Complete” problems: 3-Coloring, TSP, Knapsack, ...

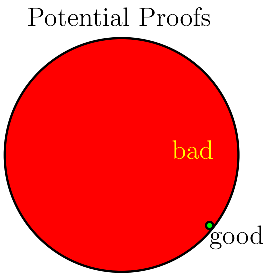
Nondeterministic Algorithm

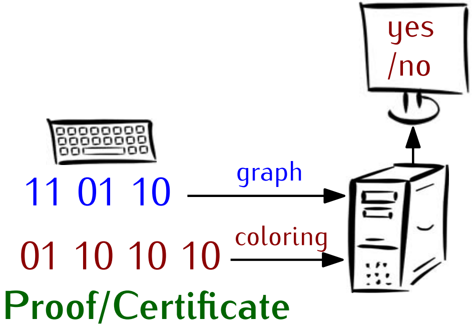




- **NP**: Nondeterministic Polynomial time
- "NP-Complete" problems: 3-Coloring, TSP, Knapsack, ...
- **$\mathbf{P} \stackrel{?}{=} \mathbf{NP}$**

Nondeterministic Algorithm





- **NP**: Nondeterministic Polynomial time
- "NP-Complete" problems: 3-Coloring, TSP, Knapsack, ...
- $\mathbf{P} \overset{?}{=} \mathbf{NP}$ $\quad$ $\mathrm{NP} \overset{?}{=} \mathrm{coNP}$

Two Big Questions

$$\mathbf{P} \stackrel{?}{=} \mathbf{NP}$$

Is finding proofs as easy as verifying them?

Is 3-coloring in Polynomial Time?

Two Big Questions

$$\mathbf{P} \overset{?}{=} \mathbf{NP}$$

Is finding proofs as easy as verifying them?

Is 3-coloring in Polynomial Time?

$$\mathbf{P} \overset{?}{=} \mathbf{BPP}$$

Does randomness $\underline{\text{truly}}$ add power?

Computational Complexity Theory

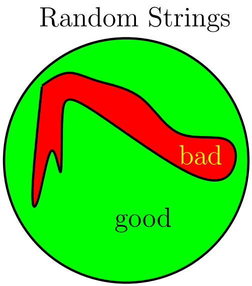
How much **time, memory space, etc.** are needed to solve problems?

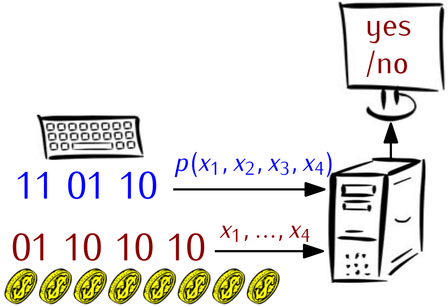
- **Is nondeterminism powerful?** P vs. NP
    - Conjecture: P$\neq$ NP
    - Techniques: hierarchy theorems, others
- **Is randomness powerful?** P vs. BPP, L vs. BPL
    - Conjecture: P=BPP, L=BPL
    - My work: derandomization, hierarchy theorems

Derandomization

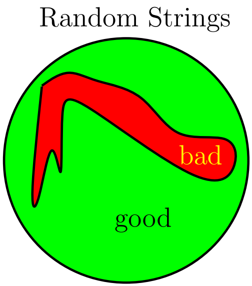
Naive Derandomization

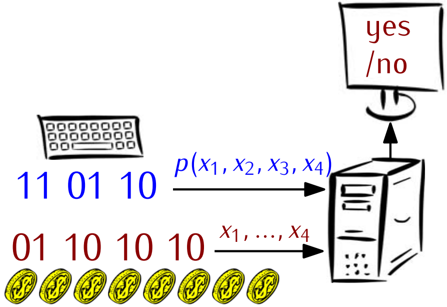
Naive Derandomization





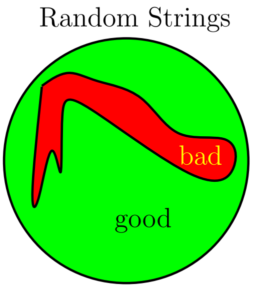
Naive Derandomization

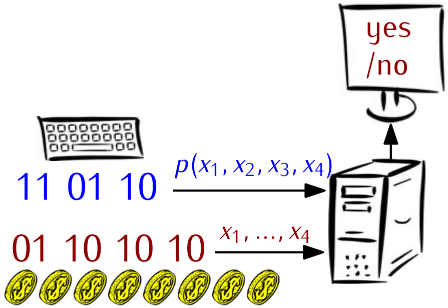




Try all possible random bit strings

Naive Derandomization

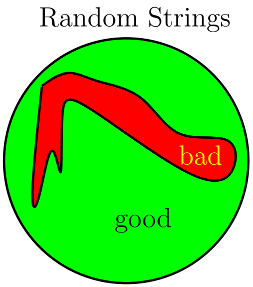


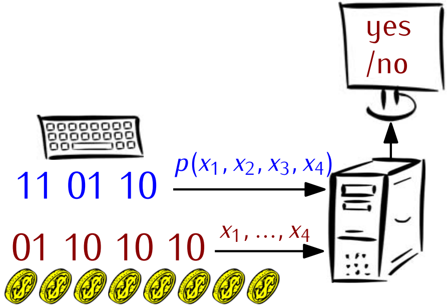


Try all possible random bit strings – exponentially many

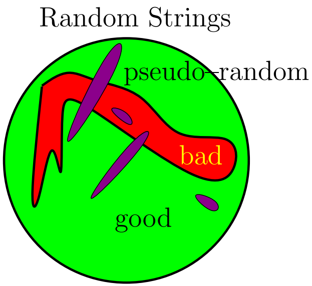
Derandomization – the Standard PRG Approach

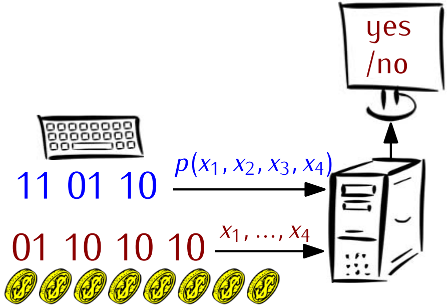
Derandomization – the Standard PRG Approach



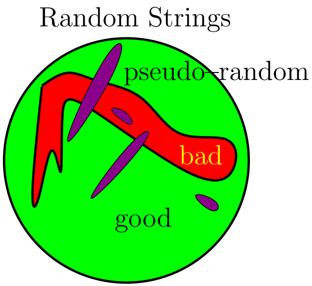


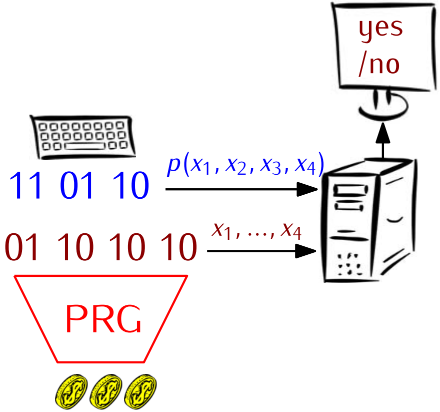
Derandomization – the Standard PRG Approach



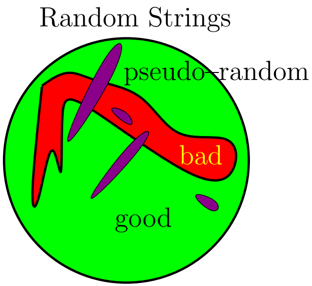


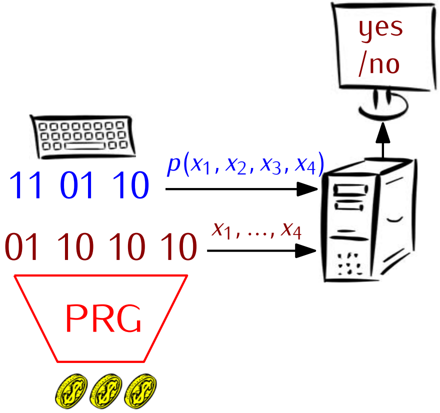
Derandomization – the Standard PRG Approach





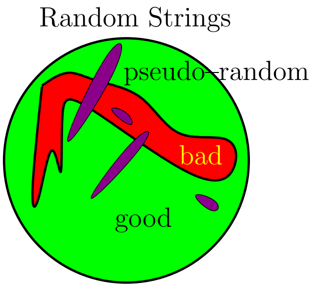
Derandomization – the Standard PRG Approach

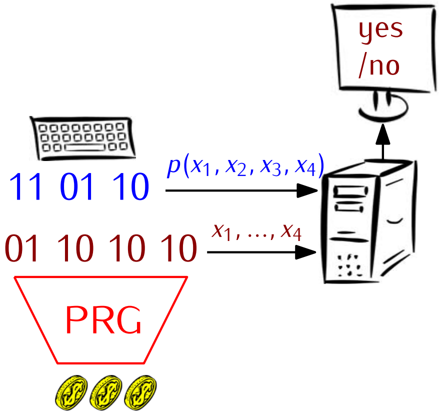




Poly many strings to try

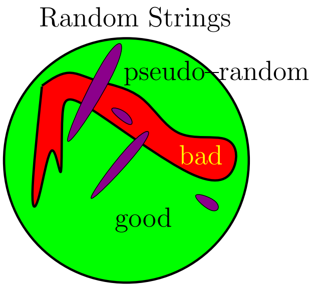
Derandomization – the Standard PRG Approach

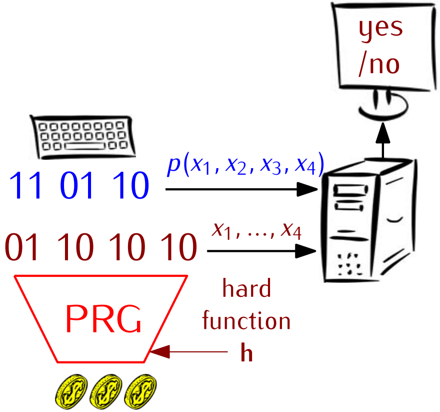




Poly many strings to try $\Rightarrow$ $O(\log n)$ seed, exp stretch

Derandomization – the Standard PRG Approach

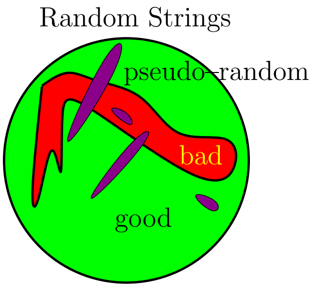


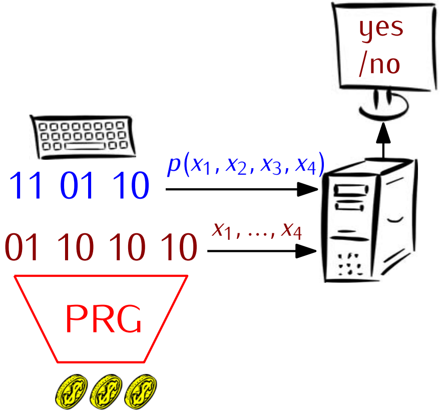


Poly many strings to try $\Rightarrow$ $O(\log n)$ seed, exp stretch

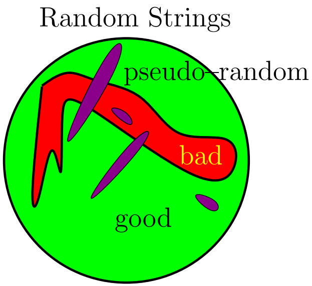
A New Approach – Typically-Correct Derandomization

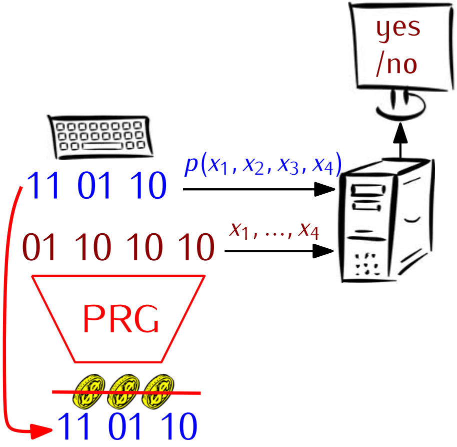
A New Approach – Typically-Correct Derandomization



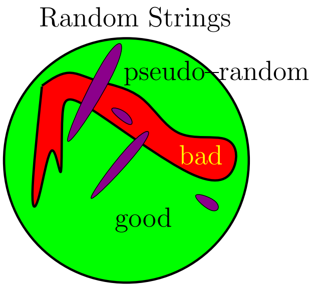


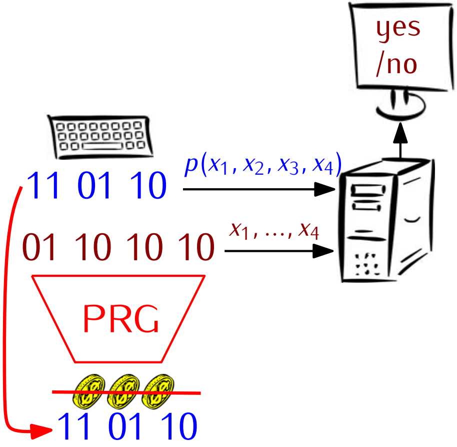
A New Approach – Typically-Correct Derandomization





A New Approach – Typically-Correct Derandomization





Seed length n, poly stretch

Standard Use of PRG's vs. Typ-Correct

Standard Use of PRG's vs. Typ-Correct

Standard Derandomization	**Typ-Correct Derandomization**
[Nisan & Wigderson, ...]	[Kinne, Van Melkebeek, Shaltiel]

Standard Use of PRG's vs. Typ-Correct

Standard Derandomization [Nisan & Wigderson, ...]	**Typ-Correct Derandomization** [Kinne, Van Melkebeek, Shaltiel]
• Always correct	• Small # mistakes

Standard Use of PRG's vs. Typ-Correct

Standard Derandomization [Nisan & Wigderson, ...]	**Typ-Correct Derandomization** [Kinne, Van Melkebeek, Shaltiel]
• Always correct	• Small # mistakes
• Run PRG many times	• Run PRG only once

Standard Use of PRG's vs. Typ-Correct

Standard Derandomization [Nisan & Wigderson, ...]	**Typ-Correct Derandomization** [Kinne, Van Melkebeek, Shaltiel]
• Always correct	• Small # mistakes
• Run PRG many times	• Run PRG only once
• Need exponential stretch	• Need only poly stretch

Standard Use of PRG's vs. Typ-Correct

Standard Derandomization [Nisan & Wigderson, ...]	**Typ-Correct Derandomization** [Kinne, Van Melkebeek, Shaltiel]
• Always correct	• Small # mistakes
• Run PRG many times	• Run PRG only once
• Need exponential stretch	• Need only poly stretch
• Conditional results	• Unconditional results: fast parallel time, streaming, communication protocols

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others
- **Is randomness powerful?** P vs. BPP, L vs. BPL
  - Conjecture: P=BPP, L=BPL
  - My work: derandomization, hierarchy theorems

Hierarchy Theorems

Hierarchy Theorems

Hierarchy Theorems

- Fix a model of computing

Hierarchy Theorems

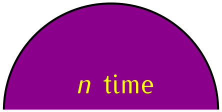
- Fix a model of computing (deterministic, randomized, nondeterministic)

Hierarchy Theorems

- Fix a model of computing
  (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?

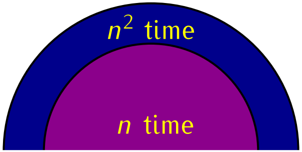
Hierarchy Theorems

- Fix a model of computing (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



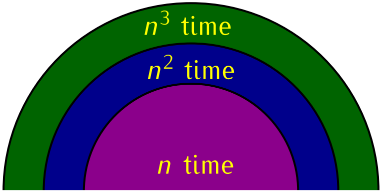
Hierarchy Theorems

- Fix a model of computing (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



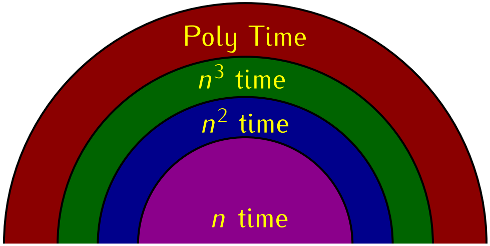
Hierarchy Theorems

- Fix a model of computing (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



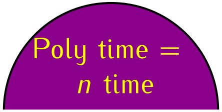
Hierarchy Theorems

- Fix a model of computing (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



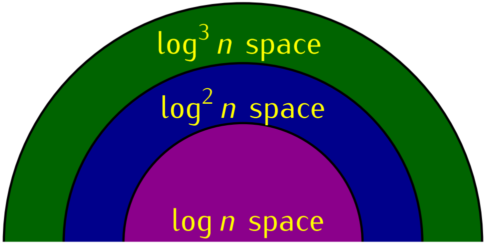
Hierarchy Theorems

- Fix a model of computing (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



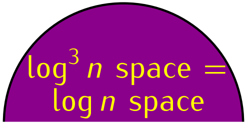
Hierarchy Theorems

- Fix a model of computing
  (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



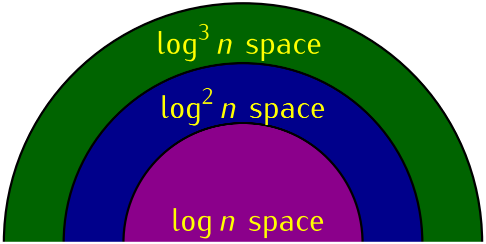
Hierarchy Theorems

- Fix a model of computing
  (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



Hierarchy Theorems

- Fix a model of computing (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?



Hierarchy Theorems

- Fix a model of computing
  (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?

Hierarchy Theorems

- Fix a model of computing
  (deterministic, randomized, nondeterministic)
- Can we achieve more given more resources?

- My work: hierarchy theorems for randomized algorithms

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others
- **Is randomness powerful?** P vs. BPP, L vs. BPL
  - Conjecture: P=BPP, L=BPL
  - My work: derandomization, hierarchy theorems

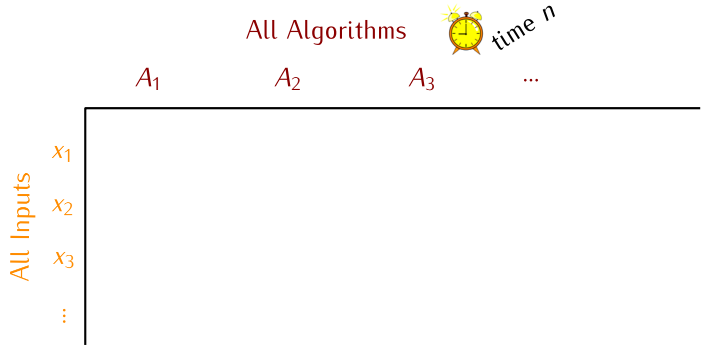
Hierarchy Theorems for Deterministic Algorithms

Hierarchy Theorems for Deterministic Algorithms

All Algorithms time n

A_1 A_2 A_3 ...

Hierarchy Theorems for Deterministic Algorithms



Hierarchy Theorems for Deterministic Algorithms

All Algorithms time n

All Inputs

$$\begin{array}{c|cccc}
 & A_1 & A_2 & A_3 & \cdots \\
\hline
x_1 & A_1(x_1) & A_2(x_1) & A_3(x_1) & \\
x_2 & A_1(x_2) & A_2(x_2) & A_3(x_2) & \cdots \\
x_3 & A_1(x_3) & A_2(x_3) & A_3(x_3) & \\
\vdots & & \vdots & &
\end{array}$$

Hierarchy Theorems for Deterministic Algorithms

All Algorithms

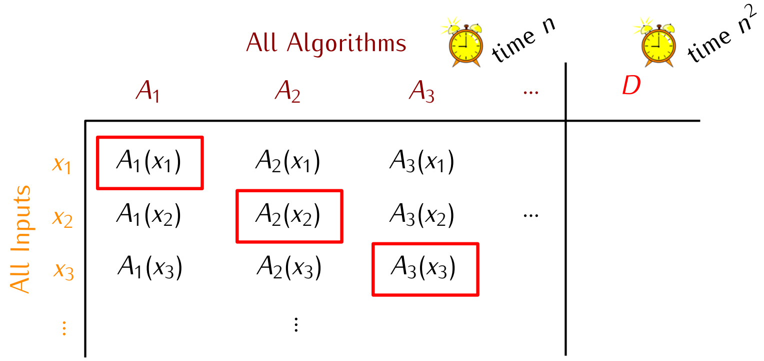
time n

time n^2

All Inputs

	A_1	A_2	A_3	$\cdots$	D
x_1	$A_1(x_1)$	$A_2(x_1)$	$A_3(x_1)$		
x_2	$A_1(x_2)$	$A_2(x_2)$	$A_3(x_2)$	$\cdots$	
x_3	$A_1(x_3)$	$A_2(x_3)$	$A_3(x_3)$		
$\vdots$		$\vdots$			

Hierarchy Theorems for Deterministic Algorithms

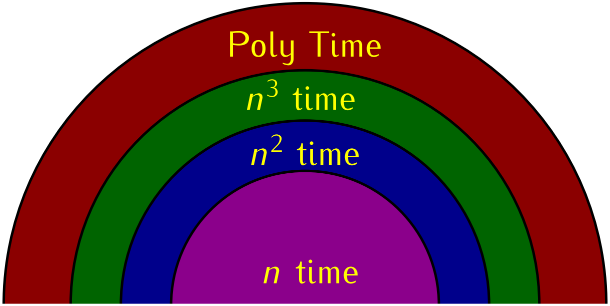


Hierarchy Theorems for Deterministic Algorithms

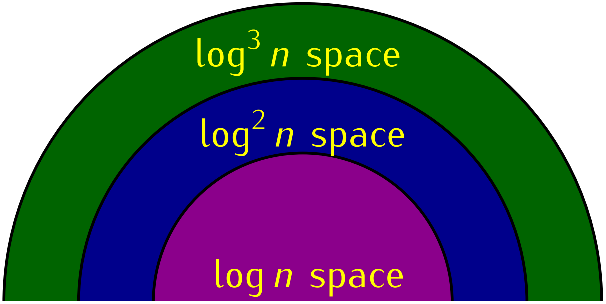
All Algorithms time n time n^2

$$\begin{array}{cc|cccc|c}
 & & A_1 & A_2 & A_3 & \dots & D \\
\hline
 & x_1 & \boxed{A_1(x_1)} & A_2(x_1) & A_3(x_1) & & \neg A_1(x_1) \\
\text{All Inputs} & x_2 & A_1(x_2) & \boxed{A_2(x_2)} & A_3(x_2) & \dots & \neg A_2(x_2) \\
 & x_3 & A_1(x_3) & A_2(x_3) & \boxed{A_3(x_3)} & & \neg A_3(x_3) \\
 & \vdots & & \vdots & & & \vdots
\end{array}$$

Hierarchy Theorems for Deterministic Algorithms

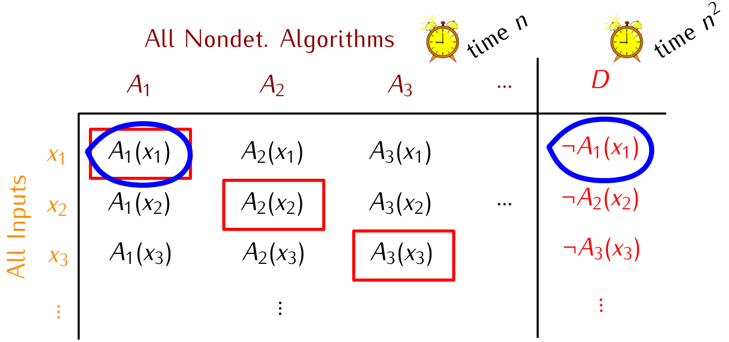


Hierarchy Theorems for Deterministic Algorithms



Hierarchy Theorems for Nondeterministic Algorithms?

Hierarchy Theorems for Nondeterministic Algorithms?



$$\begin{array}{c|cccc|c}
 & A_1 & A_2 & A_3 & \cdots & D \\
\hline
x_1 & A_1(x_1) & A_2(x_1) & A_3(x_1) & & \neg A_1(x_1) \\
x_2 & A_1(x_2) & A_2(x_2) & A_3(x_2) & \cdots & \neg A_2(x_2) \\
x_3 & A_1(x_3) & A_2(x_3) & A_3(x_3) & & \neg A_3(x_3) \\
\vdots & & \vdots & & & \vdots
\end{array}$$

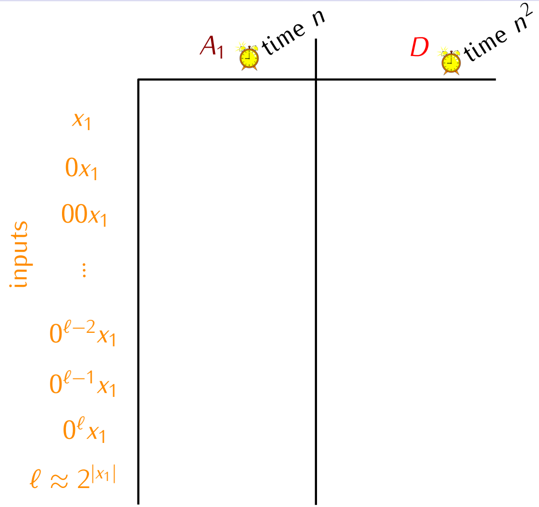
Hierarchy Theorems for Nondeterministic Algorithms

Hierarchy Theorems for Nondeterministic Algorithms

A_1 ⏰ time n

D ⏰ time n^2

Hierarchy Theorems for Nondeterministic Algorithms

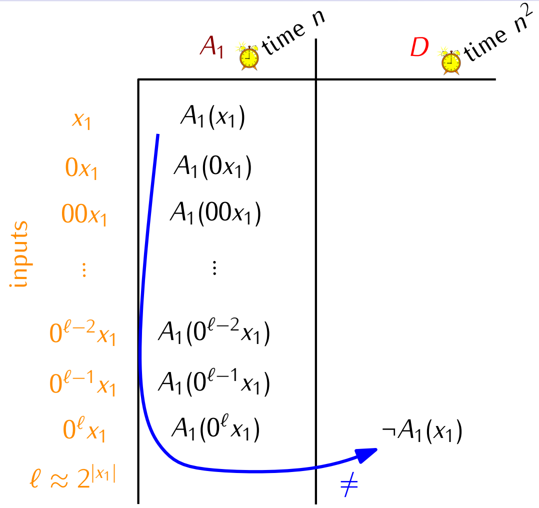


Hierarchy Theorems for Nondeterministic Algorithms

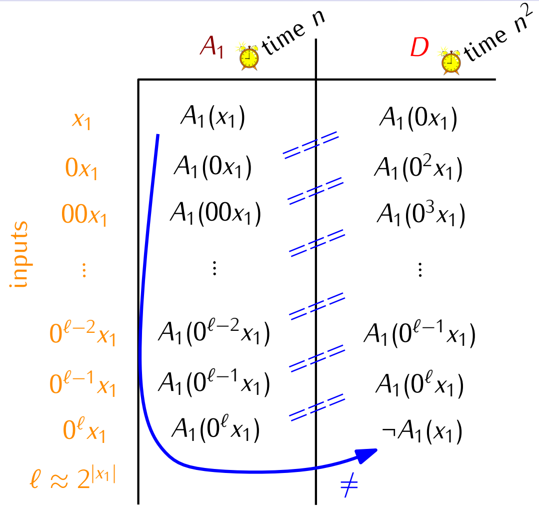
inputs

	A_1 time n	D time n^2
x_1	$A_1(x_1)$	
$0x_1$	$A_1(0x_1)$	
$00x_1$	$A_1(00x_1)$	
$\vdots$	$\vdots$	
$0^{\ell-2}x_1$	$A_1(0^{\ell-2}x_1)$	
$0^{\ell-1}x_1$	$A_1(0^{\ell-1}x_1)$	
$0^{\ell}x_1$	$A_1(0^{\ell}x_1)$	
$\ell \approx 2^{\lvert x_1 \rvert}$		

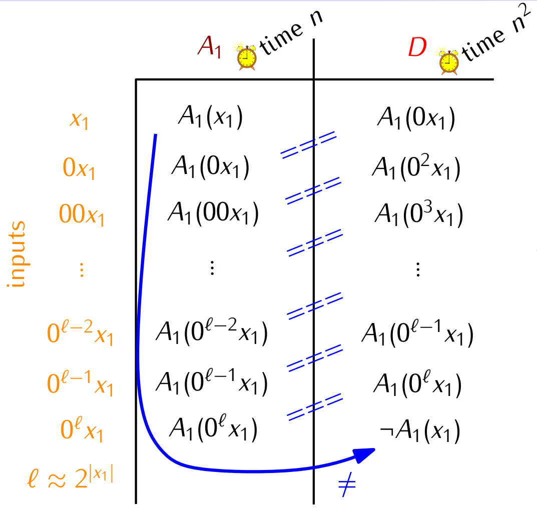
Hierarchy Theorems for Nondeterministic Algorithms



Hierarchy Theorems for Nondeterministic Algorithms

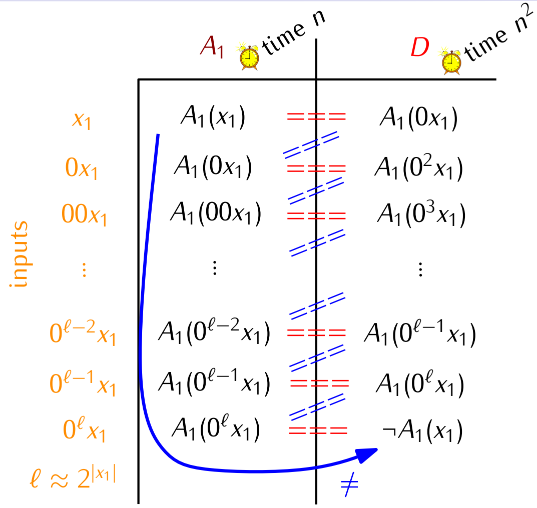


Hierarchy Theorems for Nondeterministic Algorithms



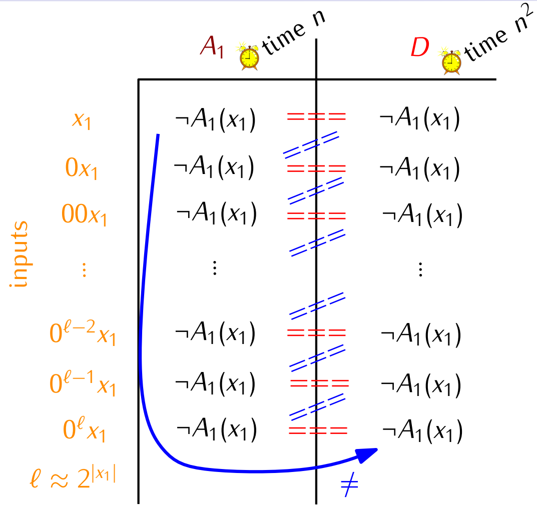
Assume A_1 the same as D on all inputs

Hierarchy Theorems for Nondeterministic Algorithms


A_1 time n
D time n^2
inputs
x_1
$0x_1$
$00x_1$
$0^{\ell-2}x_1$
$0^{\ell-1}x_1$
$0^{\ell}x_1$
$\ell \approx 2^{|x_1|}$
$A_1(x_1)$ $===$ $A_1(0x_1)$
$A_1(0x_1)$ $===$ $A_1(0^2x_1)$
$A_1(00x_1)$ $===$ $A_1(0^3x_1)$
$A_1(0^{\ell-2}x_1)$ $===$ $A_1(0^{\ell-1}x_1)$
$A_1(0^{\ell-1}x_1)$ $===$ $A_1(0^{\ell}x_1)$
$A_1(0^{\ell}x_1)$ $===$ $\neg A_1(x_1)$
$\neq$

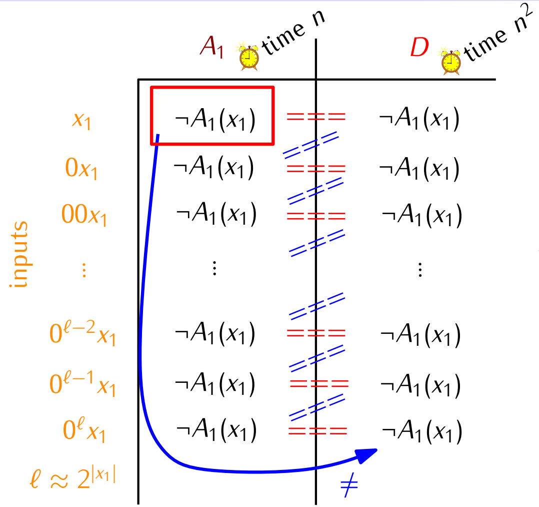
Assume A_1 the same as D on all inputs

Hierarchy Theorems for Nondeterministic Algorithms



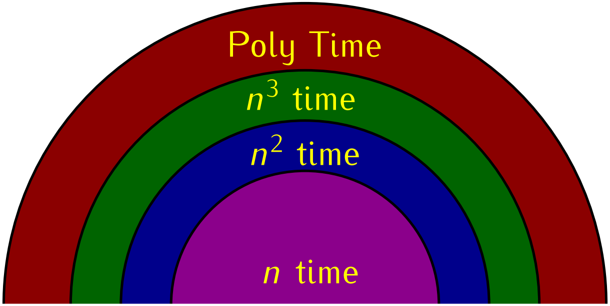
Assume A_1 the same as D on all inputs

Hierarchy Theorems for Nondeterministic Algorithms

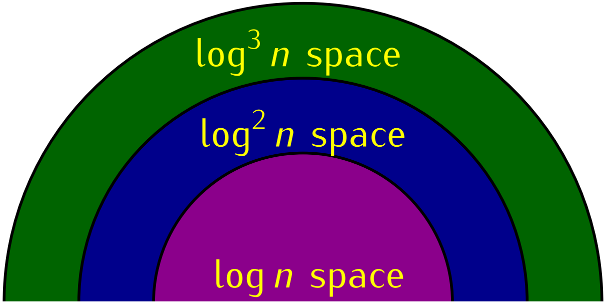


Assume A_1 the same as D on all inputs

Hierarchy Theorems for Nondeterministic Algorithms



Hierarchy Theorems for Nondeterministic Algorithms



Hierarchy Theorems for Randomized Algorithms?

Hierarchy Theorems for Randomized Algorithms?

All Randomized Algorithms time n time n^2

All Inputs

$$\begin{array}{c|cccc|c}
 & A_1 & A_2 & A_3 & \cdots & D \\
\hline
x_1 & \boxed{A_1(x_1)} & A_2(x_1) & A_3(x_1) & & \neg A_1(x_1) \\
x_2 & A_1(x_2) & \boxed{A_2(x_2)} & A_3(x_2) & \cdots & \neg A_2(x_2) \\
x_3 & A_1(x_3) & A_2(x_3) & \boxed{A_3(x_3)} & & \neg A_3(x_3) \\
\vdots & & \vdots & & & \vdots
\end{array}$$

Hierarchy Theorems for Randomized Algorithms?

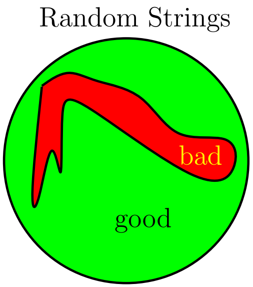
All Randomized Algorithms time n time n^2

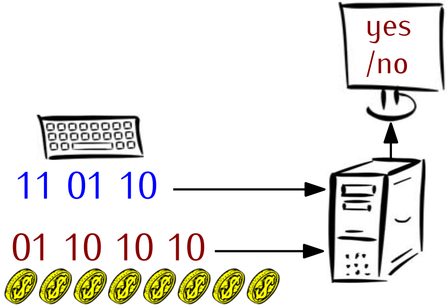
$$\begin{array}{c|cccc|c}
 & A_1 & A_2 & A_3 & \dots & D \\
\hline
x_1 & \boxed{A_1(x_1)} & A_2(x_1) & A_3(x_1) & & \neg A_1(x_1) \\
x_2 & A_1(x_2) & \boxed{A_2(x_2)} & A_3(x_2) & \dots & \neg A_2(x_2) \\
x_3 & A_1(x_3) & A_2(x_3) & \boxed{A_3(x_3)} & & \neg A_3(x_3) \\
\vdots & & \vdots & & & \vdots
\end{array}$$

All Inputs

- **What if** $\Pr[A_1(x_1) = \text{“yes”}] \approx .5$**?**

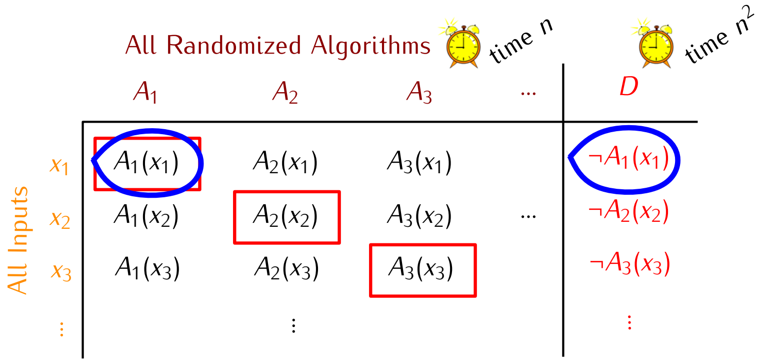
Randomized Algorithm





Bounded error: Correct with probability $> 99\%$

Hierarchy Theorems for Randomized Algorithms?


All Randomized Algorithms
time n
time n^2
A_1 A_2 A_3 ...
D
All Inputs
x_1 x_2 x_3 ⋮
$A_1(x_1)$ $A_2(x_1)$ $A_3(x_1)$
$A_1(x_2)$ $A_2(x_2)$ $A_3(x_2)$ ...
$A_1(x_3)$ $A_2(x_3)$ $A_3(x_3)$
$\neg A_1(x_1)$
$\neg A_2(x_2)$
$\neg A_3(x_3)$

- **What if** $\Pr[A_1(x_1) = \text{“yes”}] \approx .5$**?**

Hierarchy Theorems for Randomized Algorithms?

All Randomized Algorithms time n time n^2

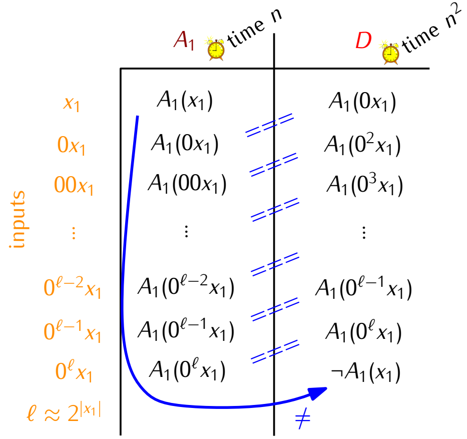
All Inputs

$$\begin{array}{c|cccc|c}
 & A_1 & A_2 & A_3 & \cdots & D \\ \hline
x_1 & \boxed{A_1(x_1)} & A_2(x_1) & A_3(x_1) & & \neg A_1(x_1) \\
x_2 & A_1(x_2) & \boxed{A_2(x_2)} & A_3(x_2) & \cdots & \neg A_2(x_2) \\
x_3 & A_1(x_3) & A_2(x_3) & \boxed{A_3(x_3)} & & \neg A_3(x_3) \\
\vdots & & \vdots & & & \vdots
\end{array}$$

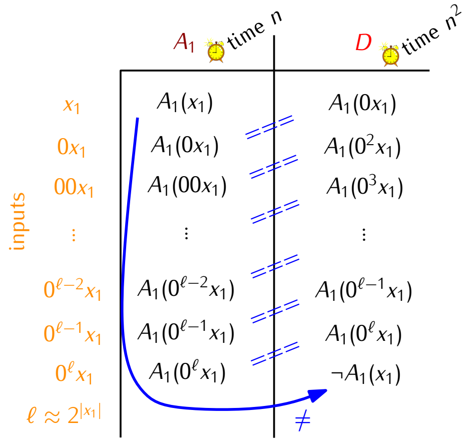
- **What if** $\Pr[A_1(x_1) =$ **"yes"**$] \approx .5$**?**
- Then D does not have bounded error, not valid

Hierarchy Theorems for Randomized Algorithms?

Hierarchy Theorems for Randomized Algorithms?

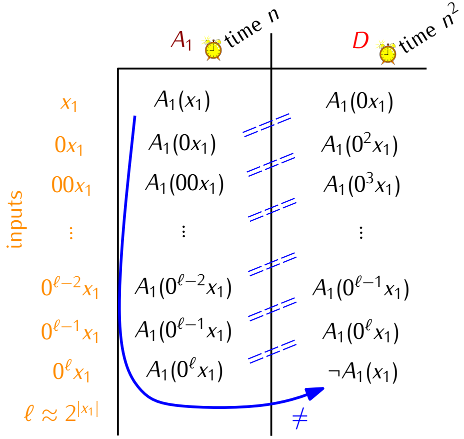


Hierarchy Theorems for Randomized Algorithms?



Make sure D has bounded error

Hierarchy Theorems for Randomized Algorithms?



Make sure D has bounded error – 1 bit of advice

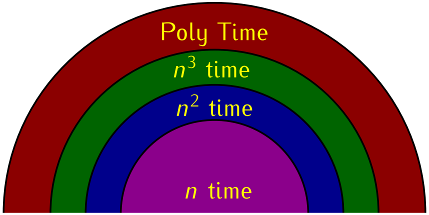
Hierarchy Theorems for Randomized Algorithms?

Hierarchy Theorems for Randomized Algorithms?

- Yes, for algorithms with 1 bit of advice!

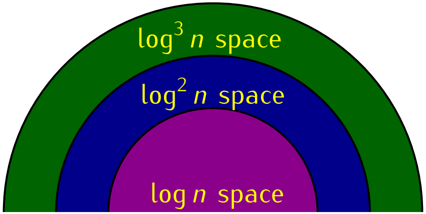
Hierarchy Theorems for Randomized Algorithms?

- Yes, for algorithms with 1 bit of advice!



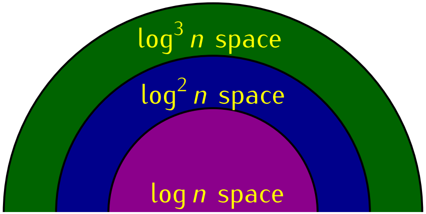
Hierarchy Theorems for Randomized Algorithms?

- Yes, for algorithms with 1 bit of advice!



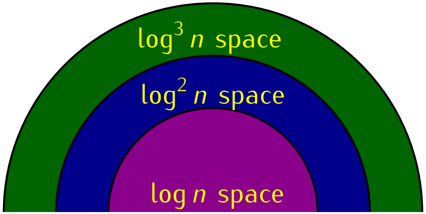
Hierarchy Theorems for Randomized Algorithms?

- Yes, for algorithms with 1 bit of advice!



Hierarchy Theorems for Randomized Algorithms?

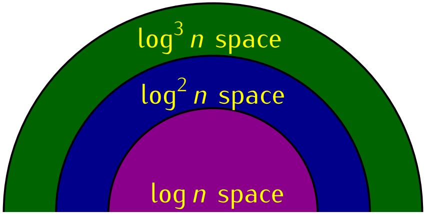
- Yes, for algorithms with 1 bit of advice!



- My work [Kinne, Van Melkebeek]

Hierarchy Theorems for Randomized Algorithms?

- Yes, for algorithms with 1 bit of advice!



- My work [Kinne, Van Melkebeek]
  Memory Space hierarchies: randomized, quantum, ...

Computational Complexity Theory

How much **time, memory space, etc.** are needed to solve problems?

- **Is nondeterminism powerful?** P vs. NP
  - Conjecture: P$\neq$ NP
  - Techniques: hierarchy theorems, others
- **Is randomness powerful?** P vs. BPP, L vs. BPL
  - Conjecture: P=BPP, L=BPL
  - My work: derandomization, hierarchy theorems

The End, Thank You!

The End, Thank You!

Slides available at:
http://www.kinnejeff.com/GoSycamores/
(or E-mail me)

More on my research (slides, papers, etc.) at:
http://www.kinnejeff.com/