Complexity Theory: NP Hard

Travelling Salesperson Problem (TSP)

By the end of this video you will be able to…

- Explain the value (and limitations) of identifying a problem is NP Hard

In TSP, given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

Brute force algorithm: Generate all paths and choose the shortest

How many permutations?

~~San Diego~~
~~Cairo~~
Johannesburg
Chennai
Lima
Paris
Beijing
Perth

How many choices for the first city? 1 (San Diego)
How many choices for the next city? 7
How many choices for the next city? 6
How many choices for the next city? 5
How many choices for the next city? 4
How many choices for the next city? 3
How many choices for the next city? 2
How many choices for the next city? 1
How many choices for the last city? 1 (San Diego)

In general we have (N-1)! permutations to try!

TSP Brute Force

N	N!
10	~3.6 million
19	1.22×10^{17} (the age of the universe)
23	# stars in the universe
59	# of atoms in the universe

Yikes!

TSP Brute Force

N	N!
10	~3.6 million
19	1.22×10^{17} (the age of the universe)
23	# stars in the universe
59	# of atoms in the universe

Yikes!

What do we do now?

TSP Brute Force

N	N!
10	~3.6 million
19	1.22×10^{17} (the age of the universe)
23	# stars in the universe
59	# of atoms in the universe

Yikes!

What do we do now?

Think really hard about a faster solution?

TSP Brute Force

N	N!
10	~3.6 million
19	1.22×10^{17} (the age of the universe)
23	# stars in the universe
59	# of atoms in the universe

Yikes!

What do we do now?

Think really hard about a faster solution?

Complexity Theory

Complexity Theory

Classifies problems by their inherent difficulty

Complexity Theory

Classifies problems by their inherent difficulty

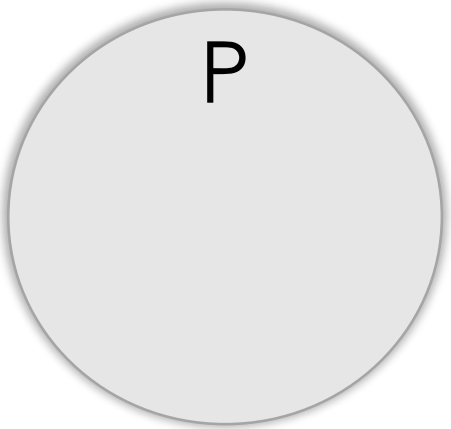
Searching a Linked List – O(n)

Sorting an Array – O(n log n)

n x n Matrix-Matrix Multiply– $O(n^{\sim 2.37})$

Complexity Theory

Classifies problems by their inherent difficulty



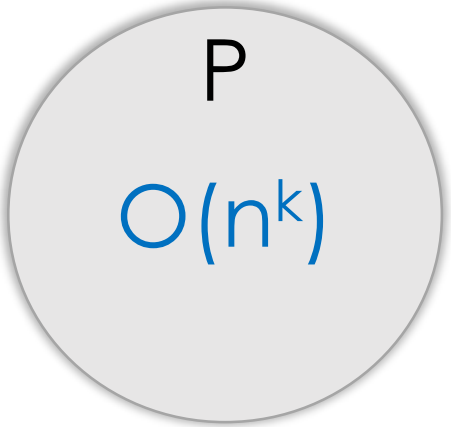
Searching a Linked List – O(n)

Sorting an Array – O(n log n)

n x n Matrix-Matrix Multiply– $O(n^{\sim 2.37})$

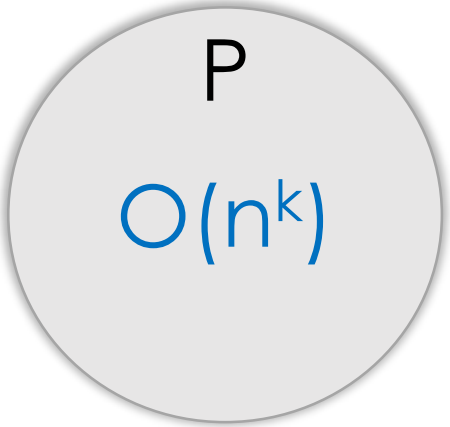
Complexity Theory

Classifies problems by their inherent difficulty



Complexity Theory

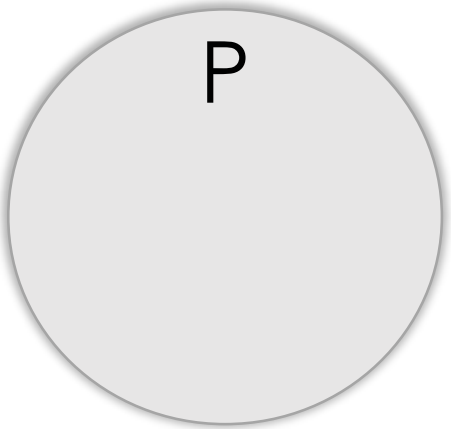
Classifies problems by their inherent difficulty



P stands for "polynomial-time"

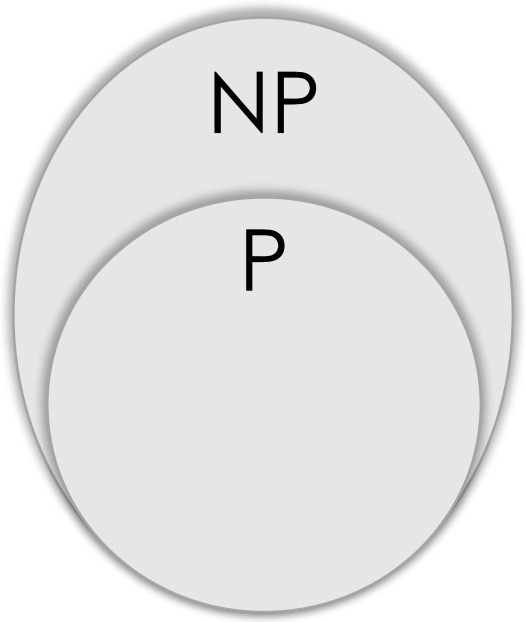
Complexity Theory

Classifies problems by their inherent difficulty



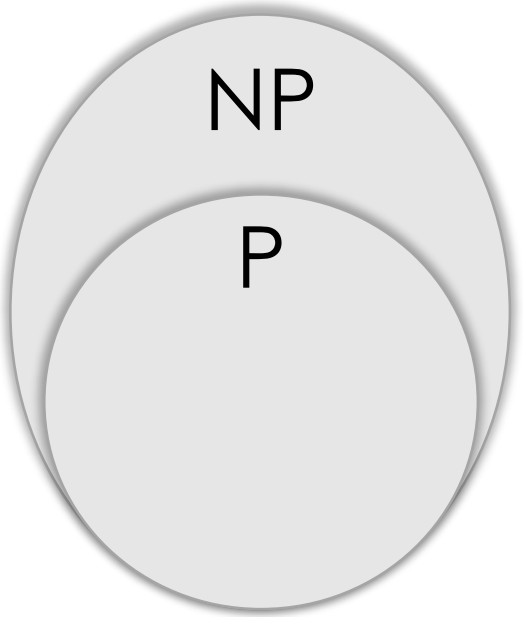
Complexity Theory

Classifies problems by their inherent difficulty



Complexity Theory

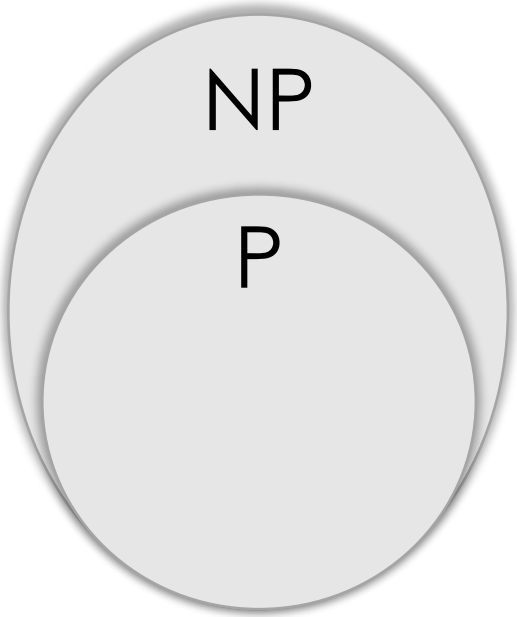
Classifies problems by their inherent difficulty



NP: Some problems seem harder to find solutions, but its still easy to verify solution correctness

Complexity Theory

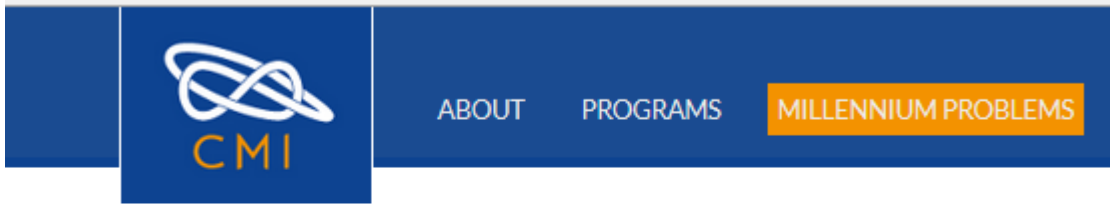
Classifies problems by their inherent difficulty



NP: Some problems seem harder to find solutions, but its still easy to verify solution correctness

NP is *believed* to contain problems harder than P

P ?= NP How to get rich and famous



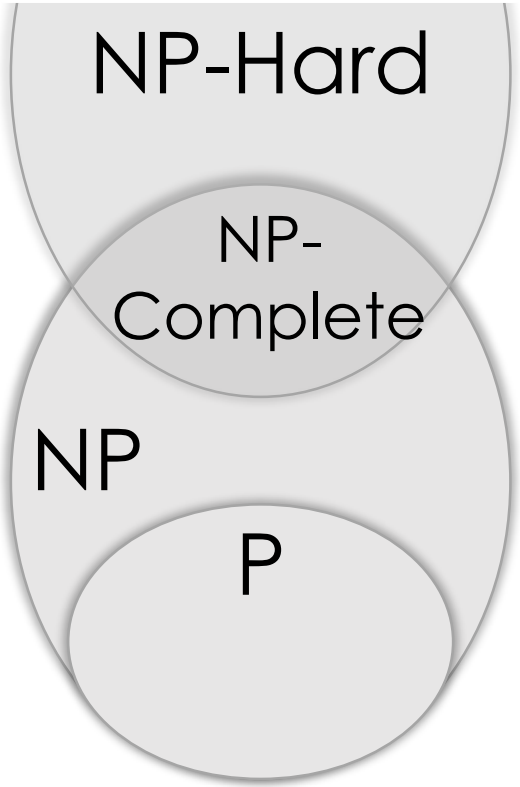
The Millennium Prize Problems

Following the decision of the Scientific Advisory Board, the Board of Directors of CMI designated a $7 million prize fund for the solutions to these problems, with $1 million allocated to the solution of each problem.

, with $1 million allocated to the solution of each problem.

http://www.claymath.org/millennium-problems

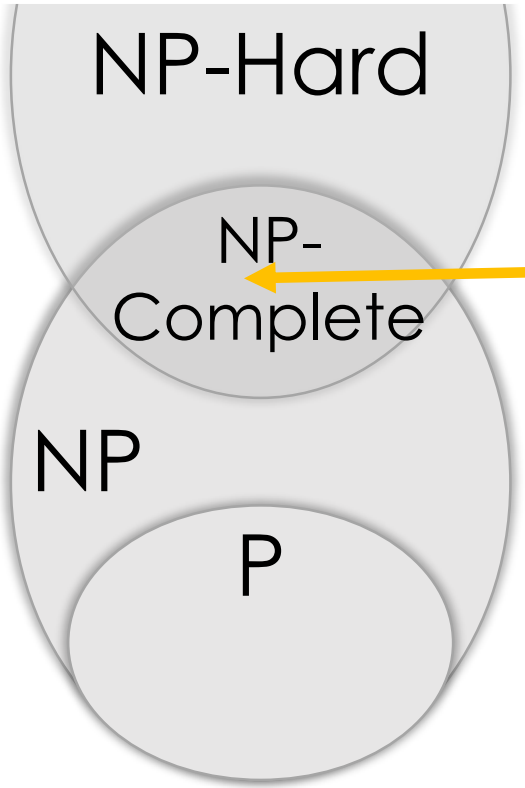
Complexity Theory



Let's expand the classifications a bit

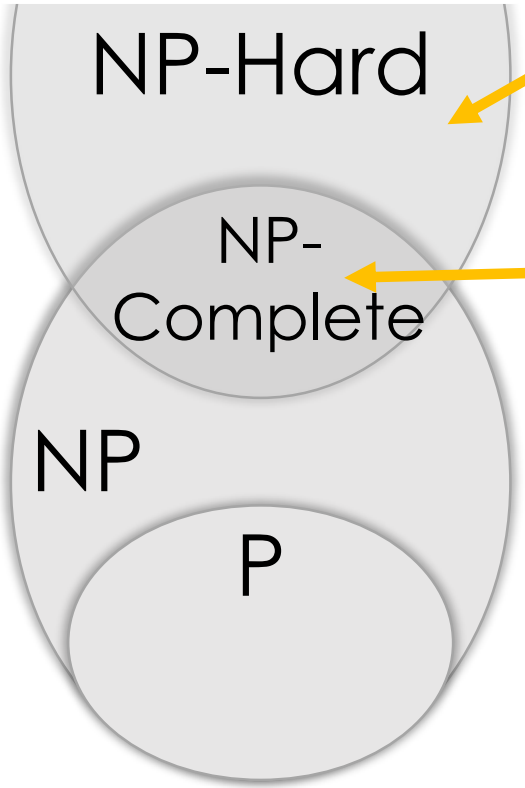
(Hierarchy if P != NP)

Complexity Theory



NP-Complete: No known polynomial time algorithm to find a solution, but can check a solution in polynomial time

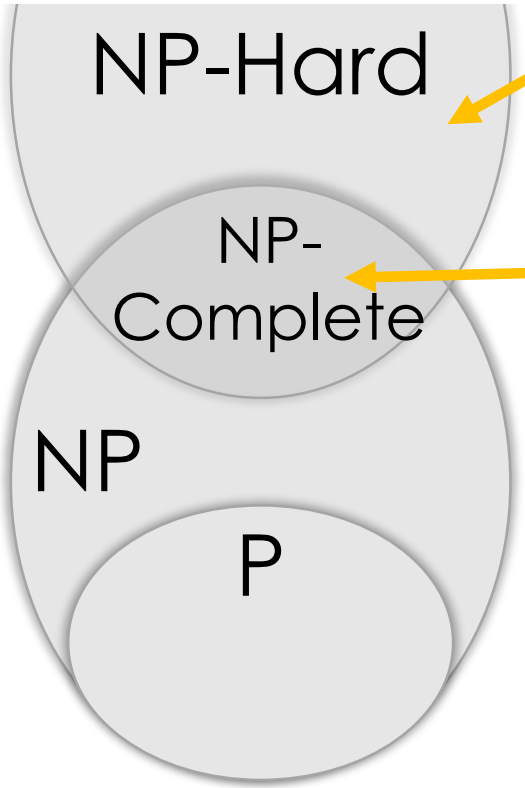
Complexity Theory



NP-Hard: Problems are *at least* as difficult to solve as hardest problems in NP

NP-Complete: No known polynomial time algorithm to find a solution, but can check a solution in polynomial time

Complexity Theory

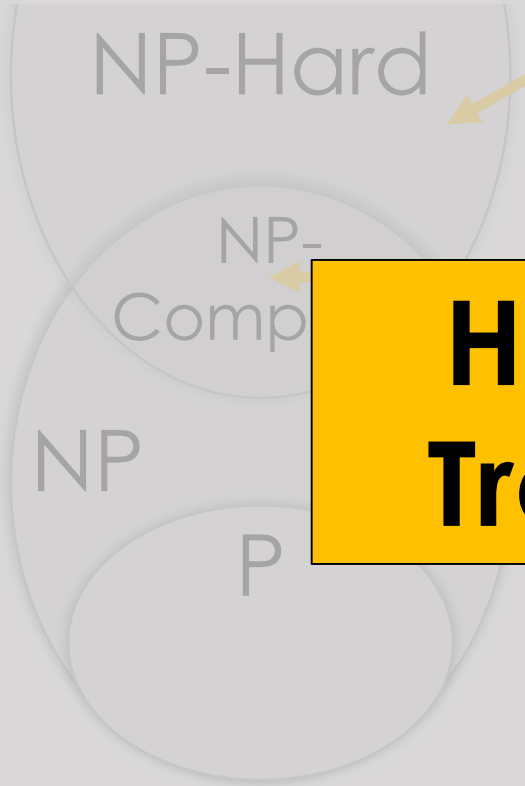


NP-Hard: Problems are *at least* as difficult to solve as hardest problems in NP

NP-Complete: No known polynomial time algorithm to find a solution, but can check a solution in polynomial time

A polynomial time solution for *any* NP-Hard problem would solve *all* NP-hard problems

Complexity Theory



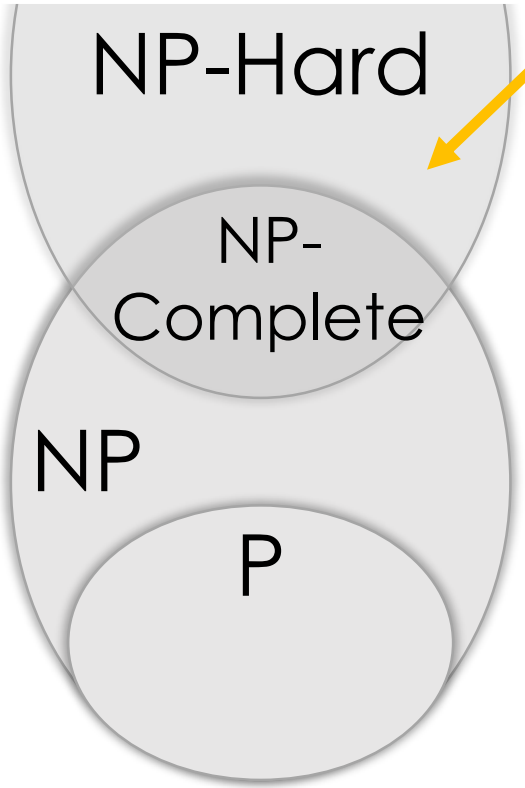
NP-Hard: Problems are *at least* as difficult to solve as hardest problems in NP

NP-Complete: No known polynomial time algorithm to find a solution, but can time

How does this relate to Travelling Salesperson?

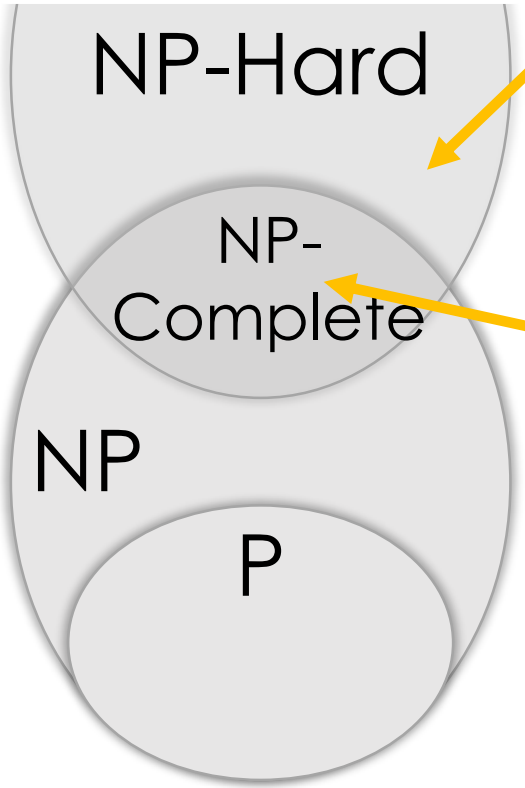
A polynomial time solution for *any* NP-Hard problem would solve *all* NP-hard problems

Complexity Theory



TSP "optimization": given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

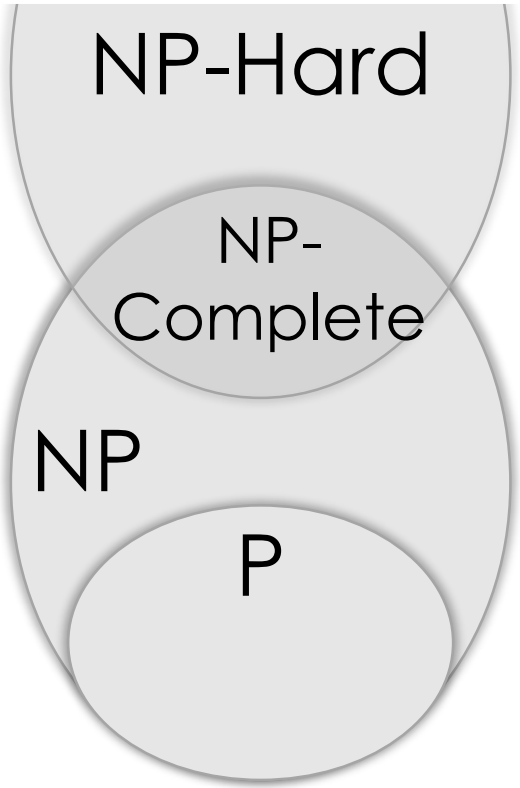
Complexity Theory



TSP "optimization": given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has minimum distance.

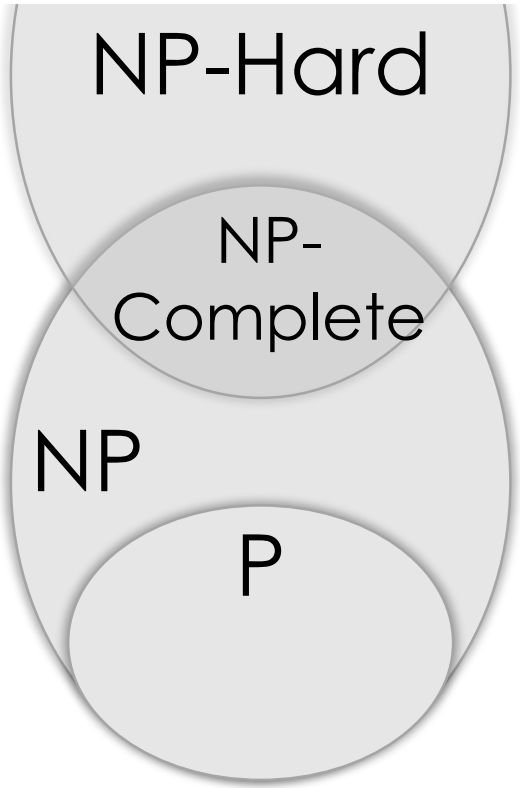
TSP "decision": given n cities with one Hometown and all pairwise distances, plan a tour starting and ending at Hometown that visits every city exactly once and has a distance less than L.

Complexity Theory



Since TSP "optimization" is NP-Hard, solving it in polynomial time may be difficult (if not impossible)

Complexity Theory



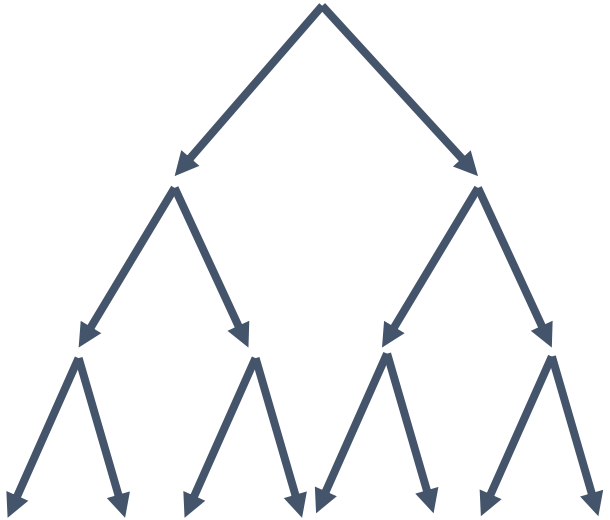
Since TSP "optimization" is NP-Hard, solving it in polynomial time may be difficult (if not impossible)

Story doesn't end here though..

BACKUP

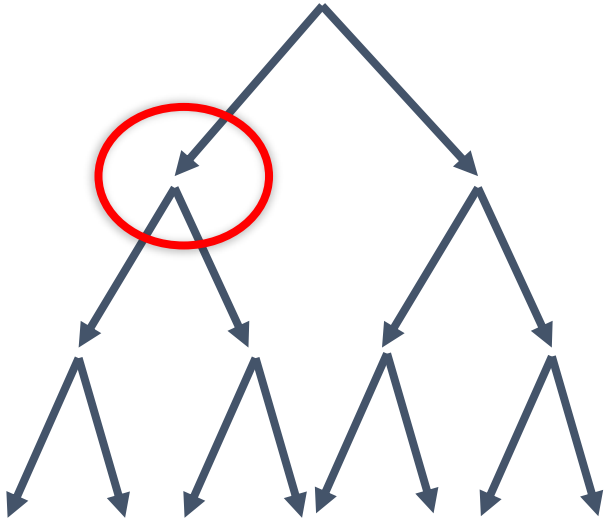
P versus NP

determinism (P)



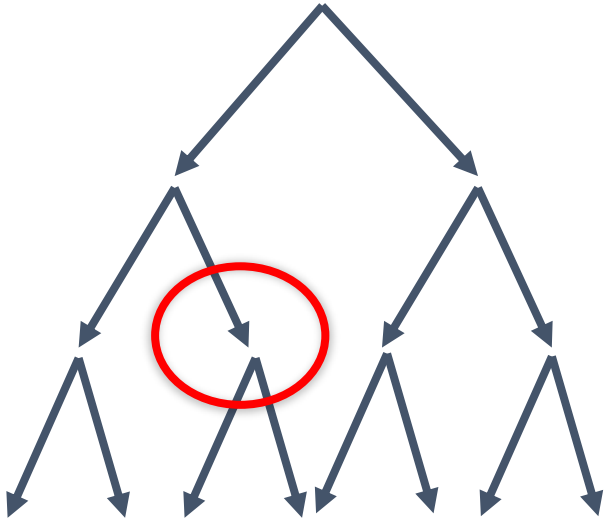
P versus NP

determinism (P)



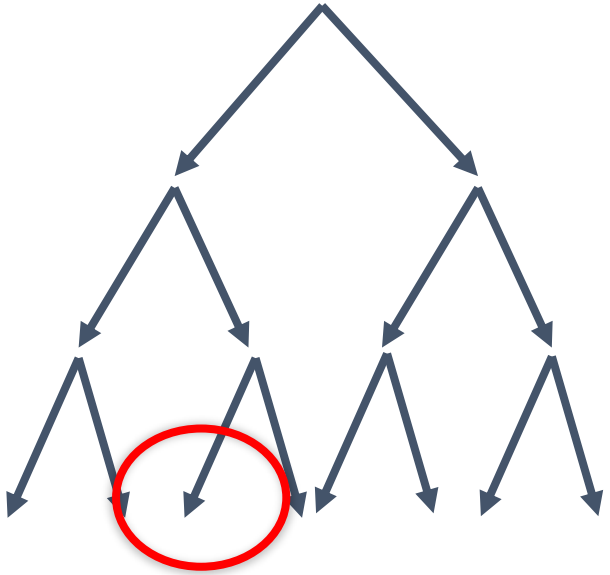
P versus NP

determinism (P)



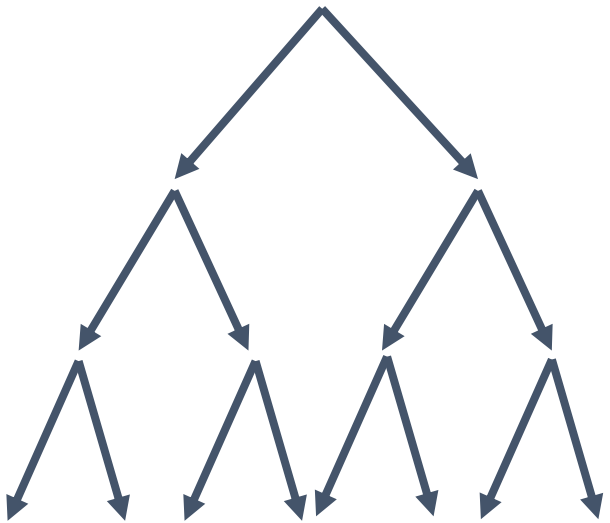
P versus NP

determinism (P)

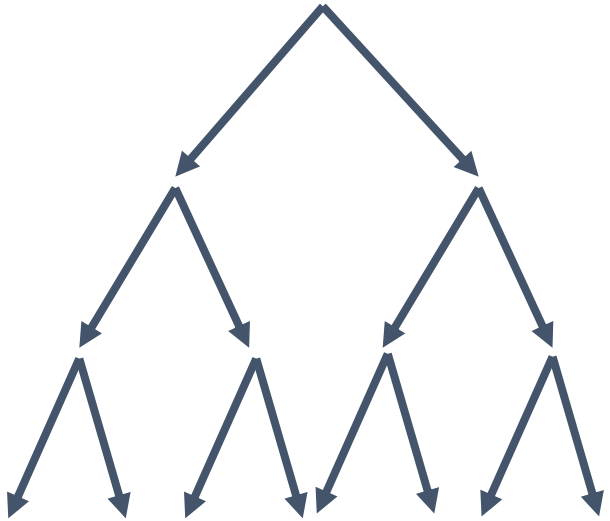


P versus NP

determinism (P)

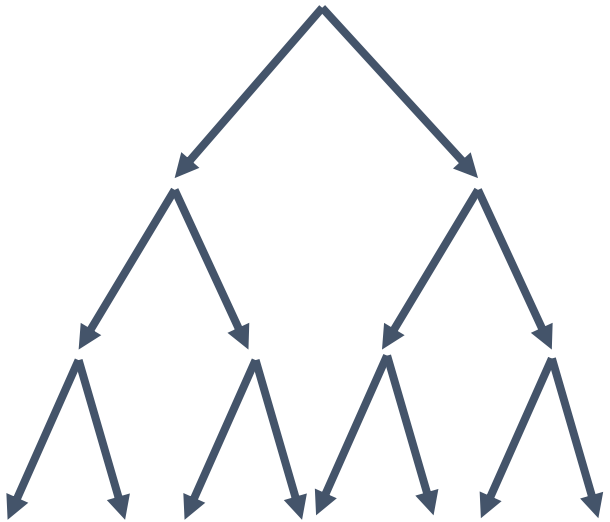


non-determinism (NP)

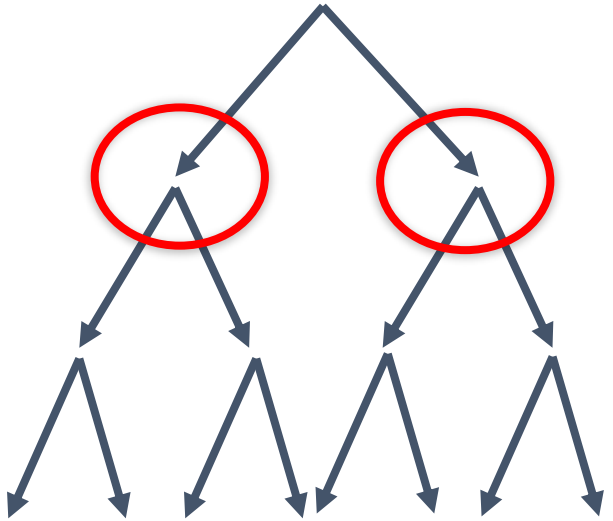


P versus NP

determinism (P)

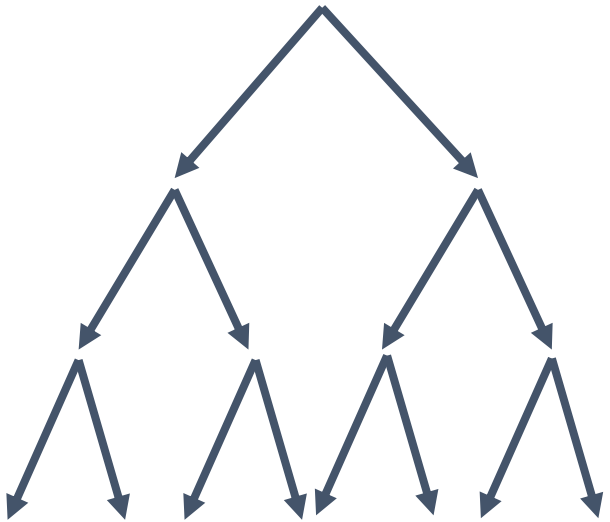


non-determinism (NP)

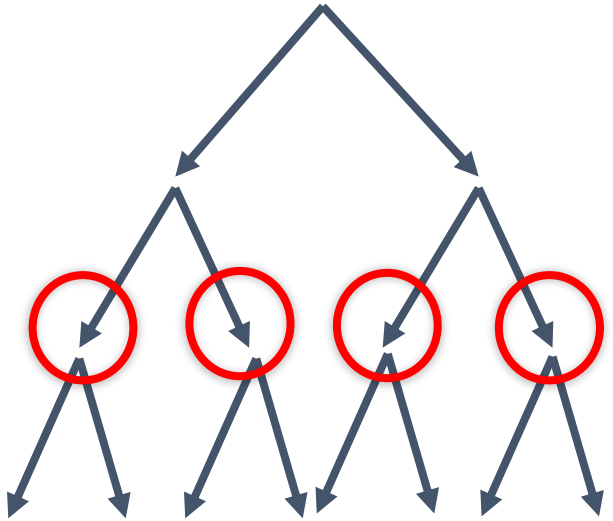


P versus NP

determinism (P)

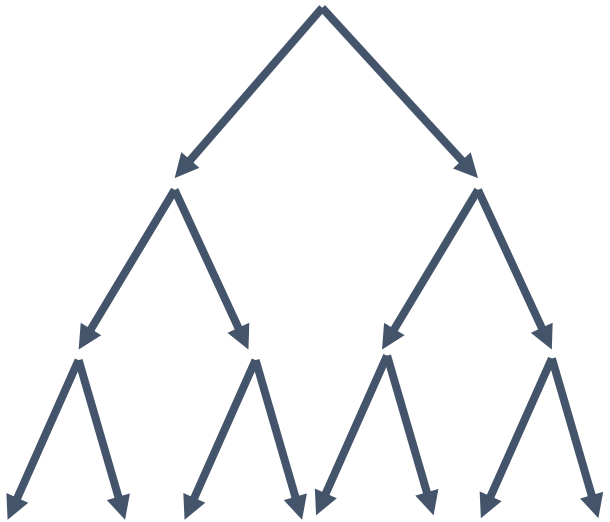


non-determinism (NP)

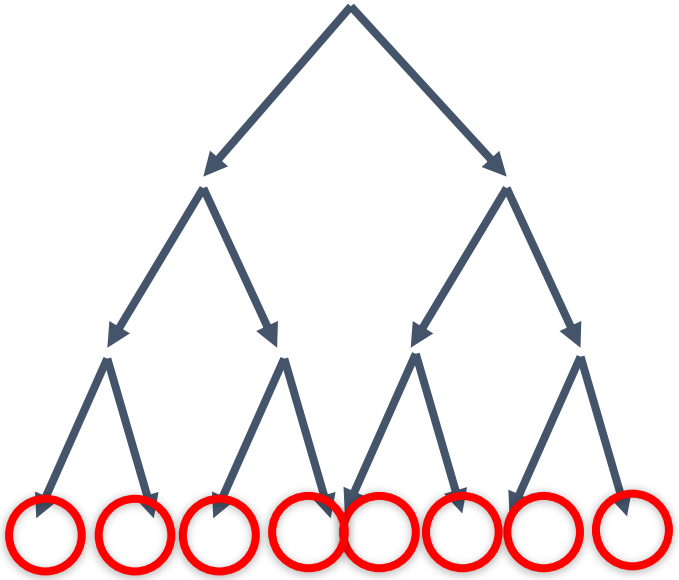


P versus NP

determinism (P)

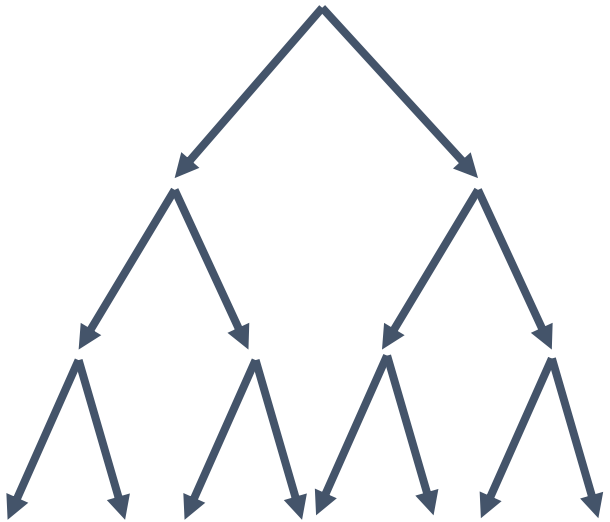


non-determinism (NP)

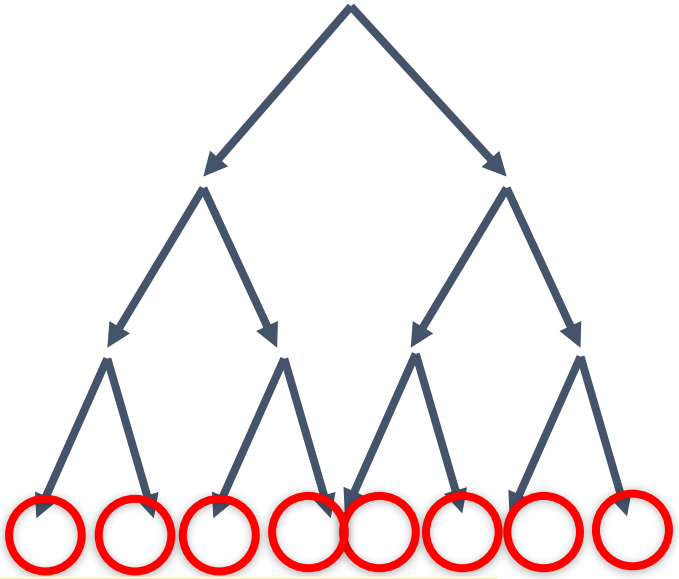


P versus NP

determinism (P)



non-determinism (NP)



Non-determinism *may* *seem* more powerful