Looking Ahead to Some Very Hard Problems

Rasmus Resen Amossen

http://rasmus.resen.org

Hard Problems

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Input Size in Bits

Input Size in Bits

Input Size in Bits

Input Size in Bits

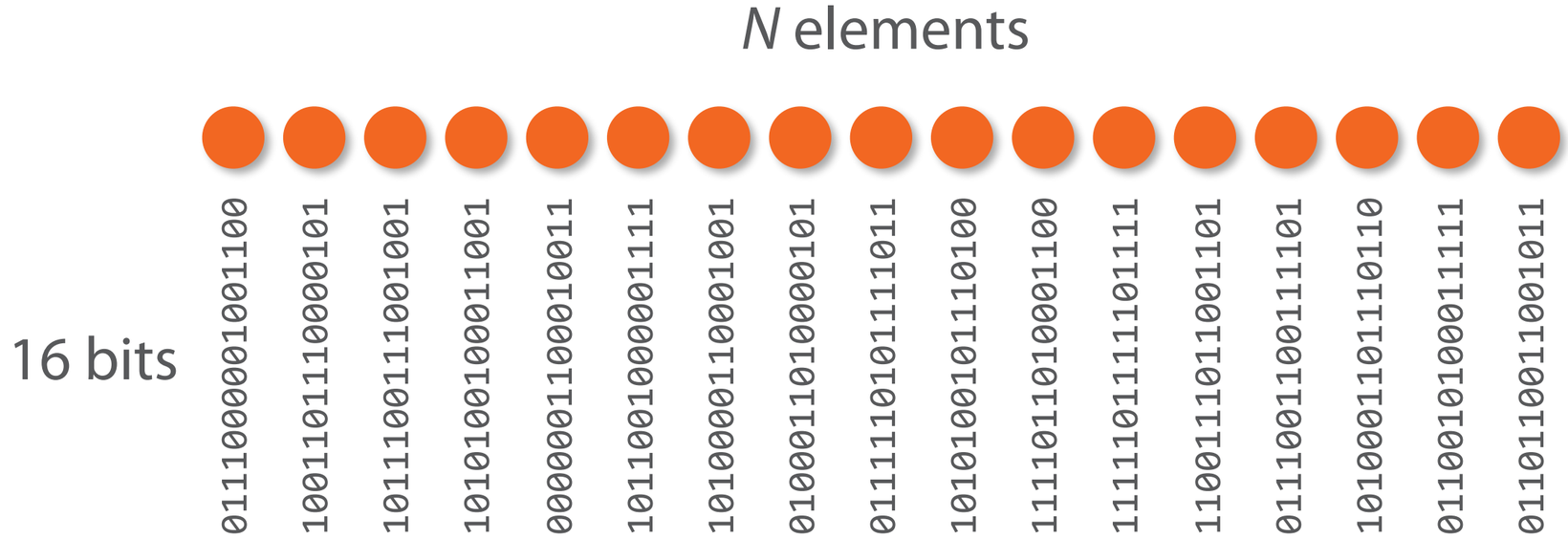
N elements

Input Size in Bits

$\Theta(N)$

Input Size in Bits

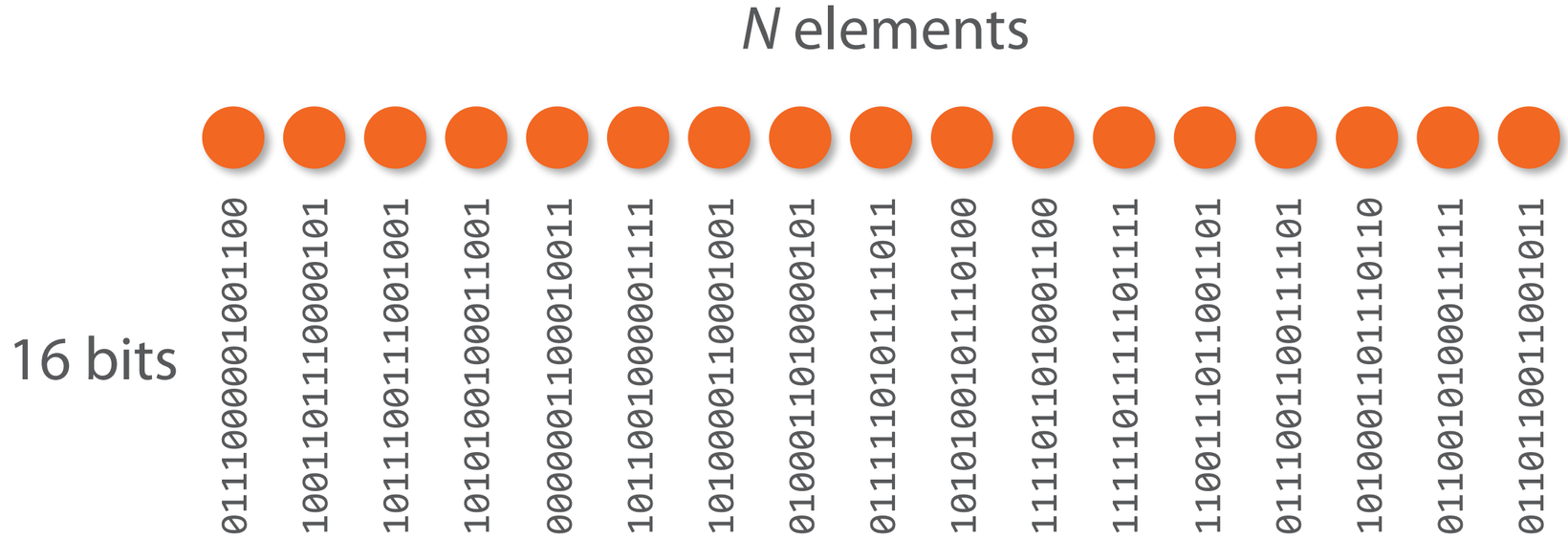
$\Theta(N)$



Input Size in Bits

$\Theta(N)$

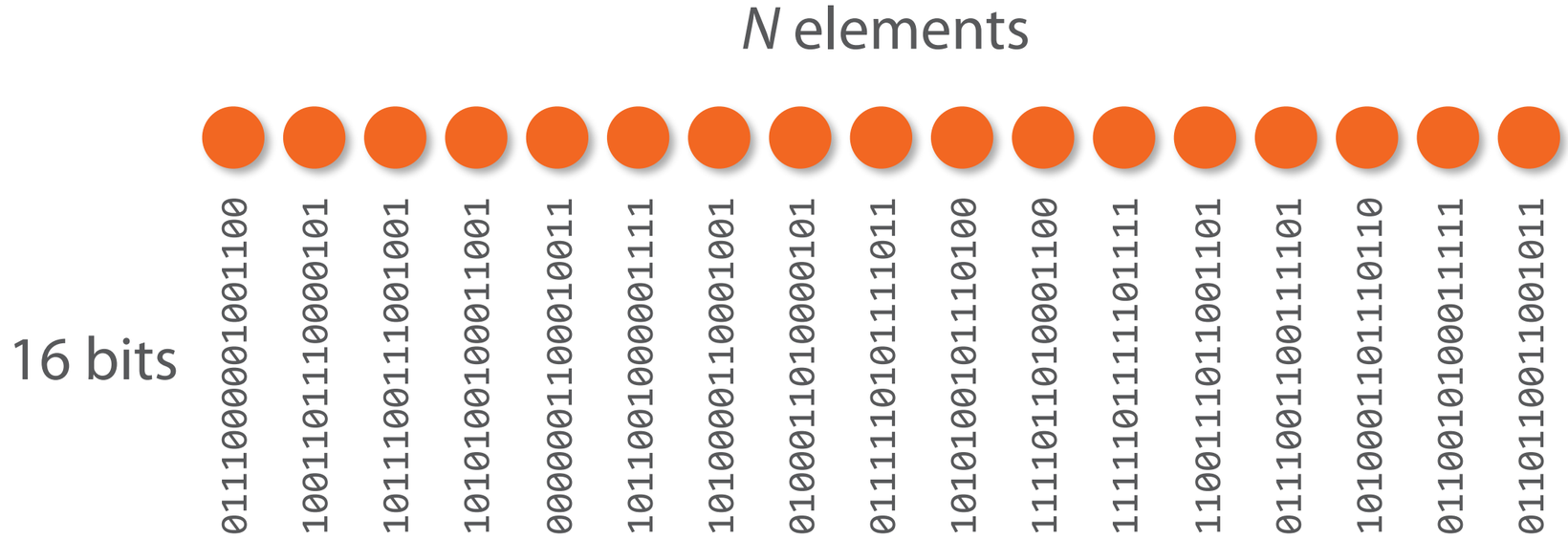
$\Theta(16 \cdot N)$



Input Size in Bits

$\Theta(N)$

$\Theta(16 \cdot N) = \Theta(N)$



Input Size in Bits

N elements

Input Size in Bits

Sorting…

N elements

Input Size in Bits

Sorting…

$O(N \cdot \log_2 N)$

N elements

Input Size in Bits

Sorting…

$O(N \cdot \log_2 N)$

Input size

N elements

Input Size in Bits

Sorting…

$O(N \cdot \log_2 N)$

Input size

#Halvings

N elements

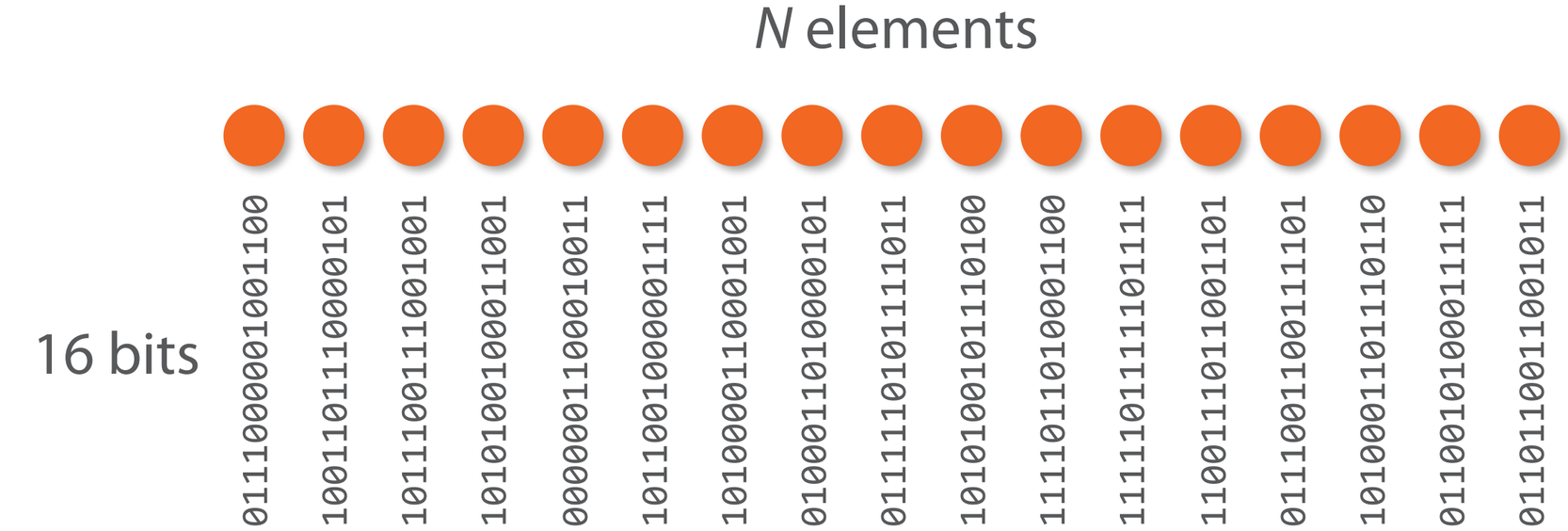
Input Size in Bits

Sorting…

$O(N \cdot \log_2 N)$

Input size

#Halvings



Input Size in Bits

Sorting…

$O(N \cdot \log_2 N)$

Input size

#Halvings

$O(16N$

16 bits

N elements

Input Size in Bits

Sorting…

$O(N \cdot \log_2 N)$

Input size

#Halvings

16 bits

N elements

Input Size in Bits

Sorting…

$O(N \cdot \log_2 N)$

Input size

#Halvings

N elements

16 bits

$O(16N \cdot \log_2 N)$

$= O(N \cdot \log_2 N)$

Input Size in Bits

Knapsack capacity: 4

Weight: 2
Value: 3

Weight: 2
Value: 1

Weight:1
Value: 3

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For *N* items and capacity of *C*:

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4



Weight: 2
Value: 3

Weight: 2
Value: 1

Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits or $2 \cdot 32 \cdot N$ bits

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits or $2 \cdot 32 \cdot N$ bits

Capacity: c bits gives a max capacity of 2^c

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits or $2 \cdot 32 \cdot N$ bits

Capacity: c bits gives a max capacity of 2^c

$\Theta(32 \cdot N$

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits or $2 \cdot 32 \cdot N$ bits

Capacity: c bits gives a max capacity of 2^c

$\Theta(32 \cdot N \cdot 2^c)$

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits or $2 \cdot 32 \cdot N$ bits

Capacity: c bits gives a max capacity of 2^c

$\Theta(32 \cdot N \cdot 2^c) = \Theta(N \cdot 2^c)$

		Capacity				
		0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits or $2 \cdot 32 \cdot N$ bits

Capacity: c bits gives a max capacity of 2^c

$\Theta(32 \cdot N \cdot 2^c) = \Theta(N \cdot 2^c)$ Ouch!

		Capacity 0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Input Size in Bits

Knapsack capacity: 4


Weight: 2
Value: 3


Weight: 2
Value: 1


Weight:1
Value: 3

For N items and capacity of C:

Dynamic programming: $\Theta(N \cdot C)$

Items: $32 \cdot N$ bits or $2 \cdot 32 \cdot N$ bits

Capacity: c bits gives a max capacity of 2^c

$\Theta(32 \cdot N \cdot 2^c) = \Theta(N \cdot 2^c)$ Ouch!

		Capacity 0	1	2	3	4
No items	i_0	0	0	0	0	0
	i_1	0	0✗	3✓	3✓	3✓
	i_2	0	0✗	3✗	3✗	4✓
	i_3	0	3✓	3✓	6✓	6✓

Pseudo polynomial

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #b

P vs NP

Heuristics
and
approximation

P vs NP

P vs NP

Polynomial

$5N^5 + 3N^4 + N$

P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$

P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$

P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$

P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$

2^N

Not a polynomial

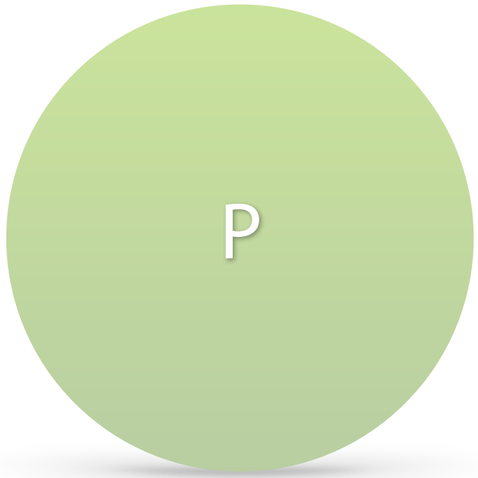
P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$



2^N

Not a polynomial

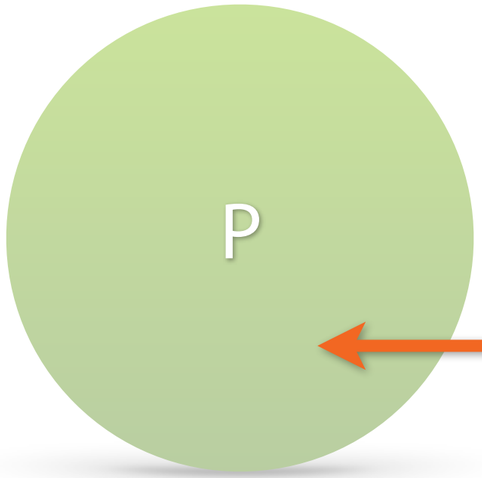
P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$



Solvable in polynomial time

2^N

Not a polynomial

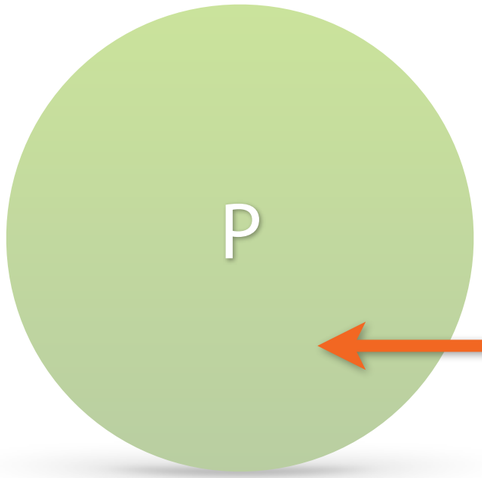
P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$



Solvable in polynomial time

Sorting

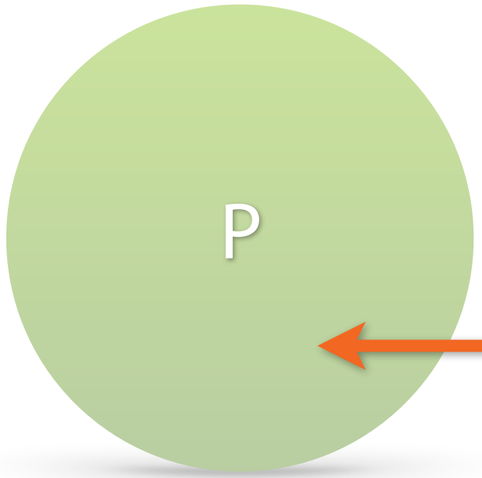
P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$



Solvable in polynomial time

Sorting

Traversing lists

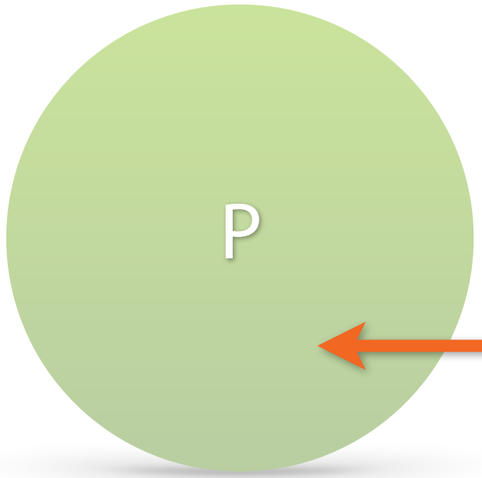
P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$



Solvable in polynomial time

Sorting

Traversing lists

Look-ups in hash tables

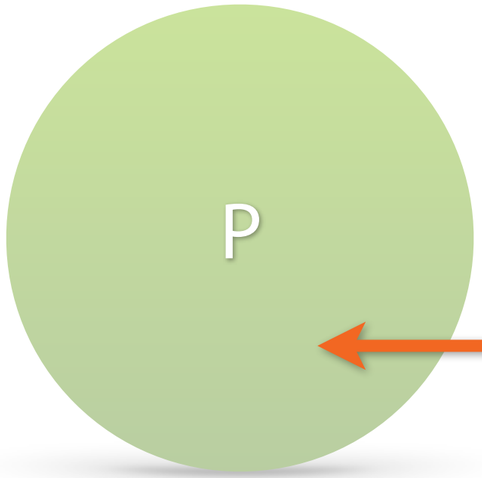
P vs NP

Polynomial

$5N^5 + 3N^4 + N$ is $O(N^5)$

$5N^3 + 3N^2 + N + 7$ is $O(N^3)$

$\log_2 N$ is $O(N)$



Solvable in polynomial time

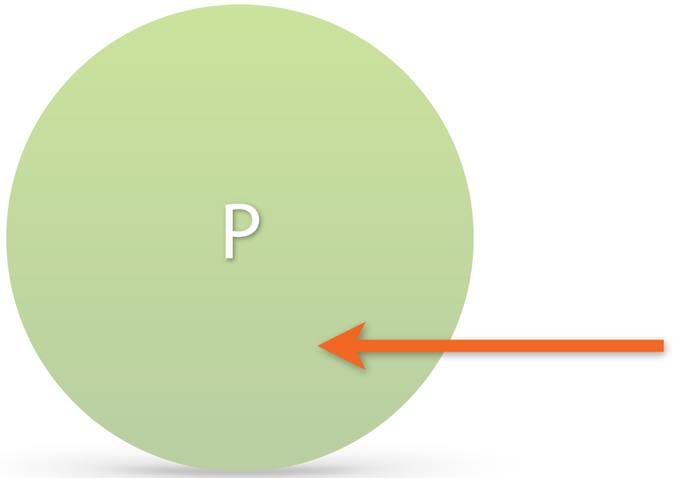
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



Solvable in polynomial time

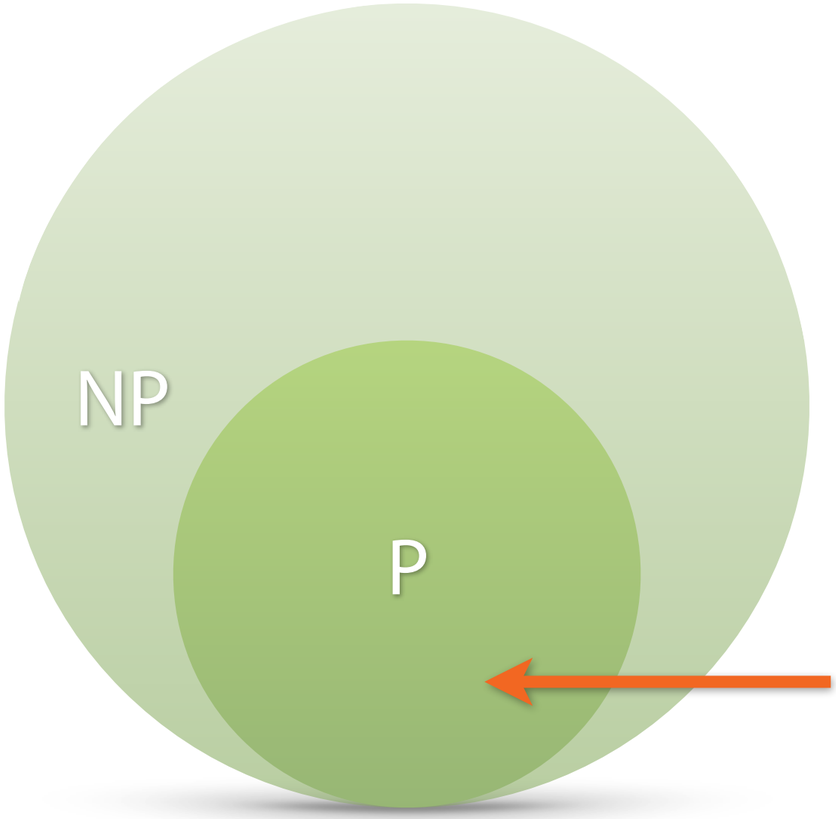
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



Solvable in polynomial time

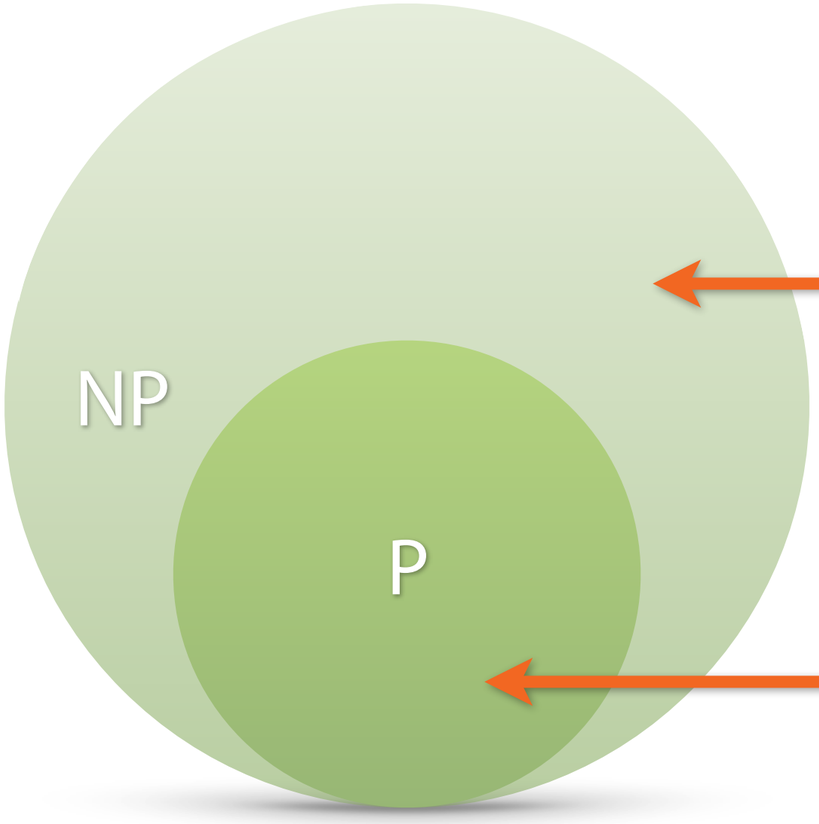
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Sorting

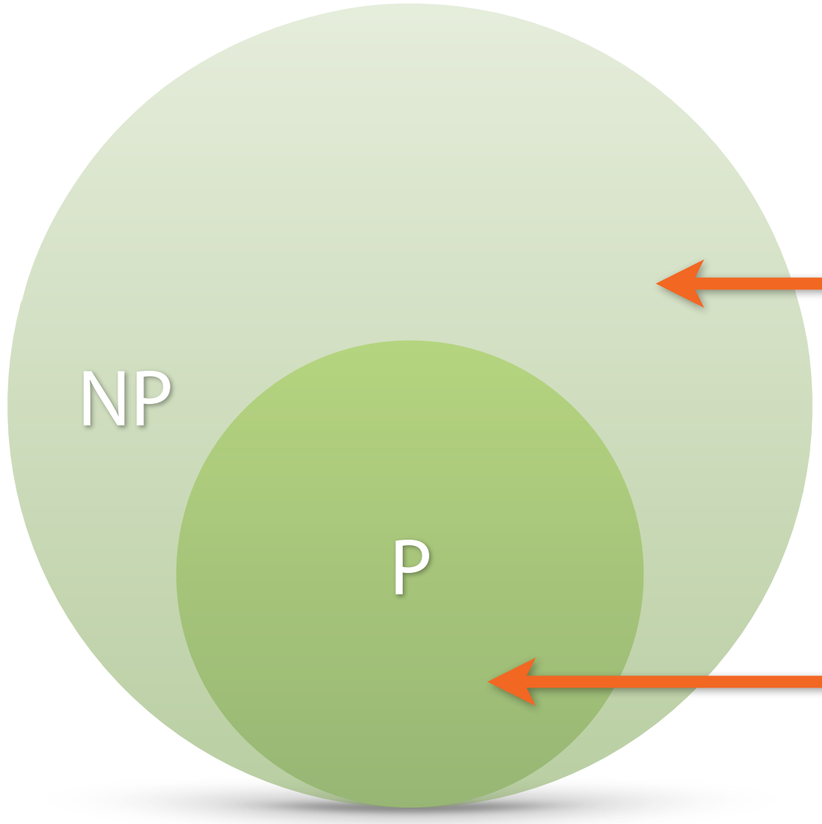
Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP

? Yes
No



Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Sorting

Finding shortest route

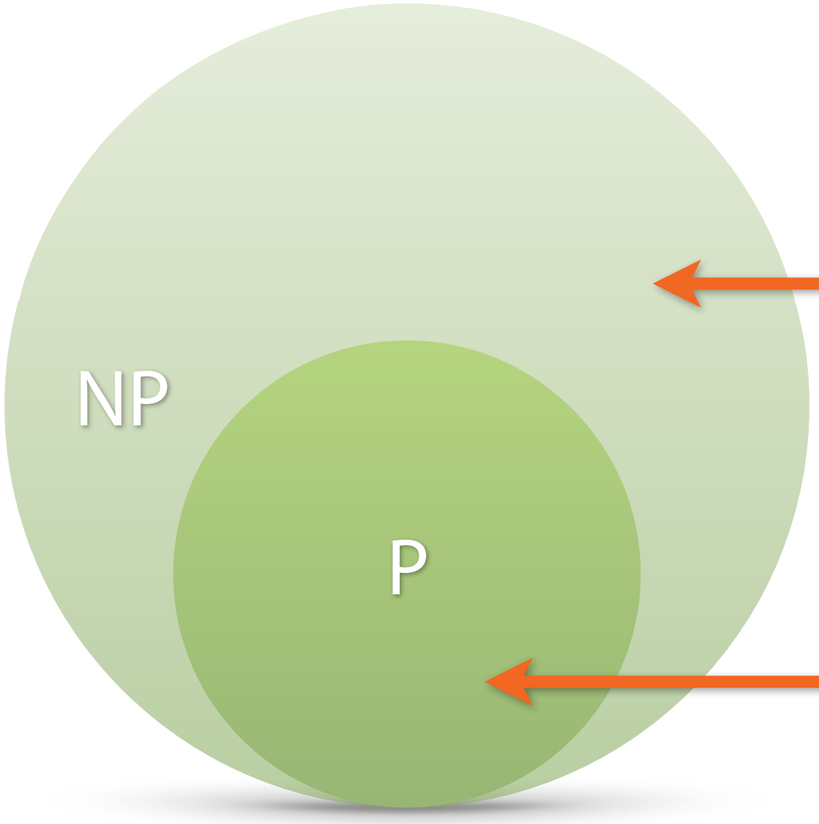
Traversing lists

Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes
No



Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Sorting

Finding shortest route

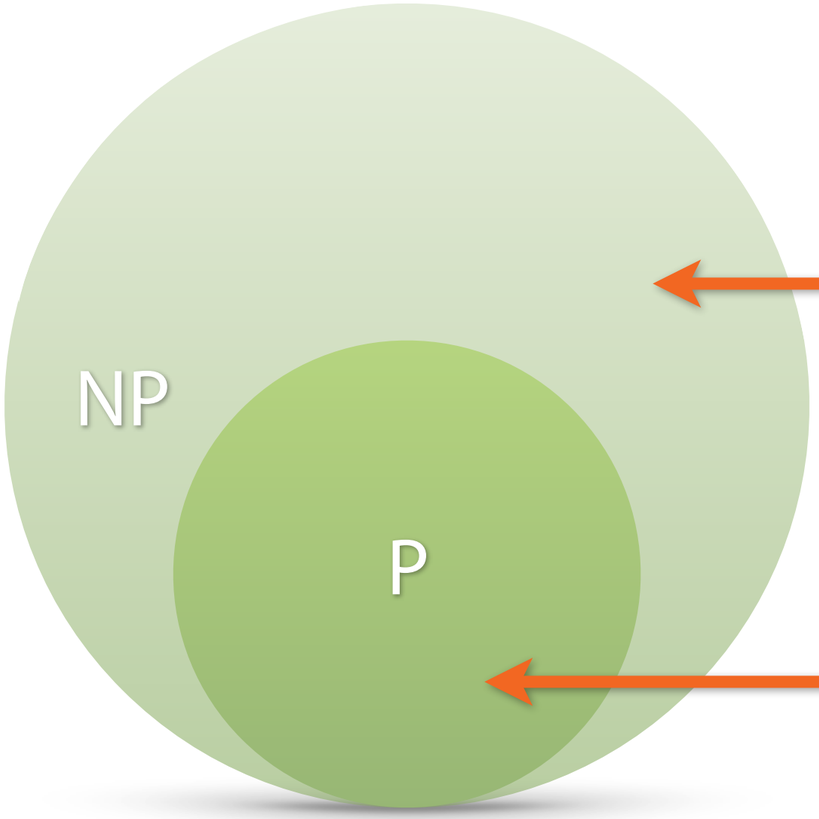
Traversing lists

Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes
No



Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Sorting

Finding shortest route

Traversing lists

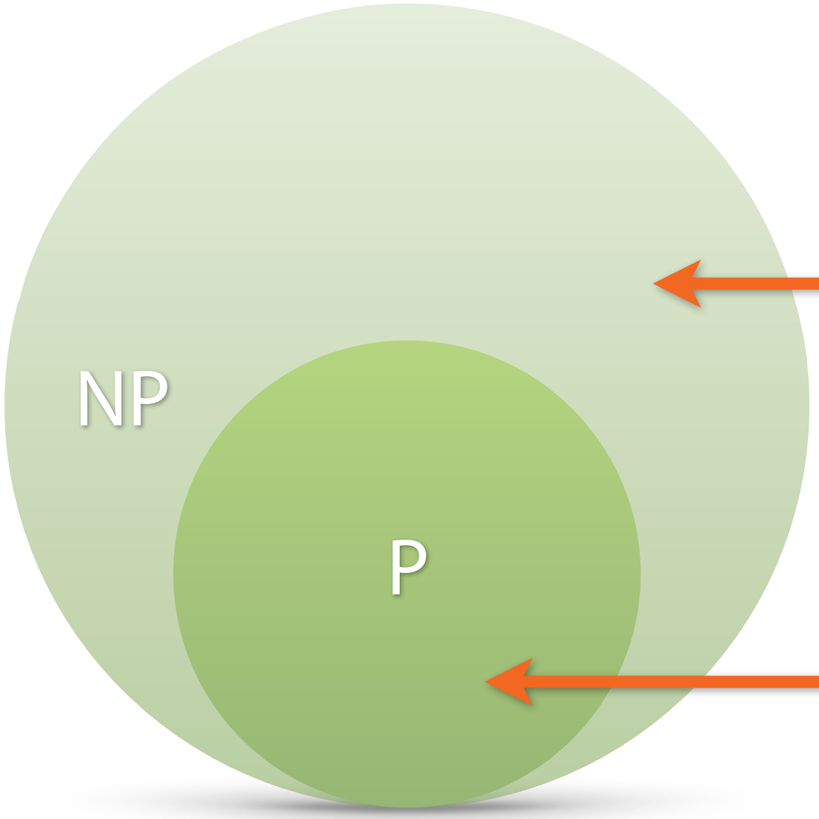
Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes / No

Certificate: a concrete route



Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Sorting

Finding shortest route

Traversing lists

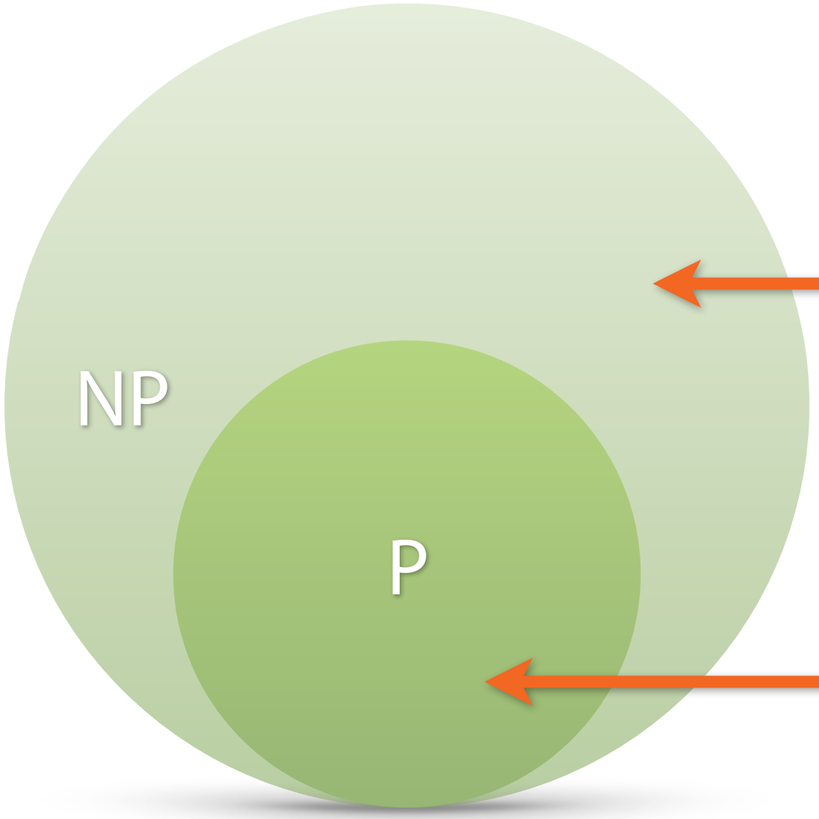
Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes → Indeed
No ↗

Certificate: a concrete route



Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Sorting

Finding shortest route

Traversing lists

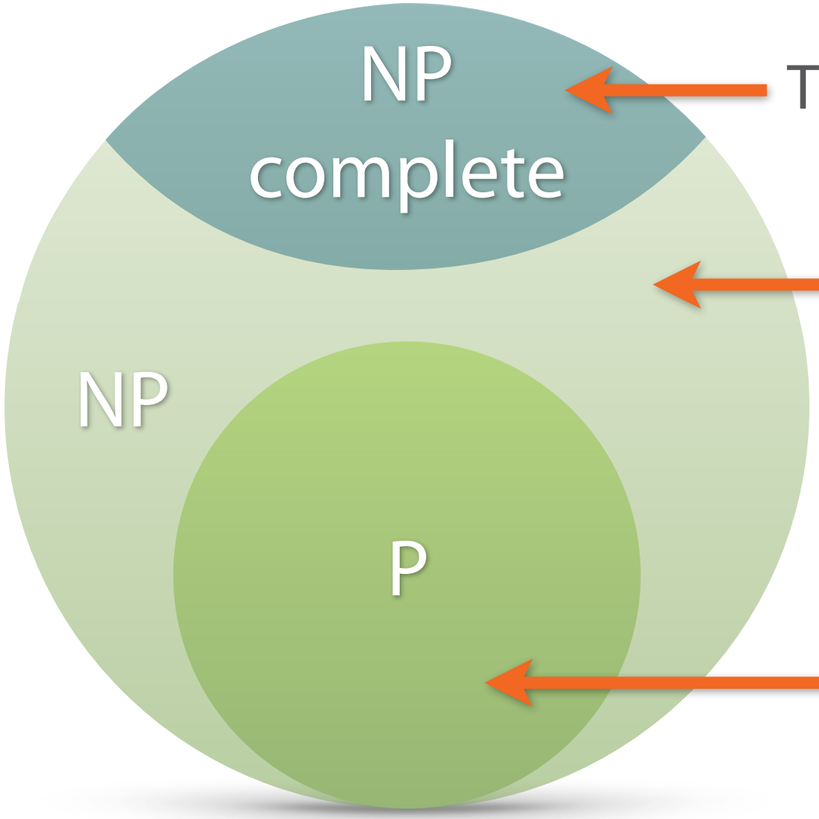
Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes → Indeed
No ↗

Certificate: a concrete route



The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Sorting

Finding shortest route

Traversing lists

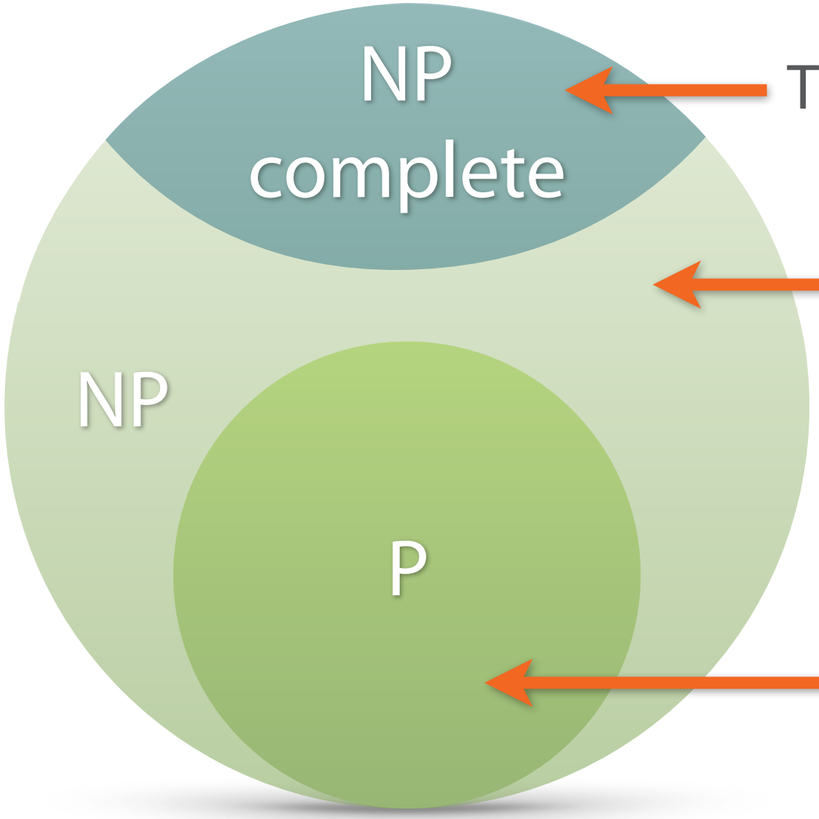
Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes → Indeed
No ↗

Certificate: a concrete route



The hardest in NP

Subset sum

Decision problems ***verifiable*** in polynomial time

Sorting

Finding shortest route

Solvable in polynomial time

Traversing lists

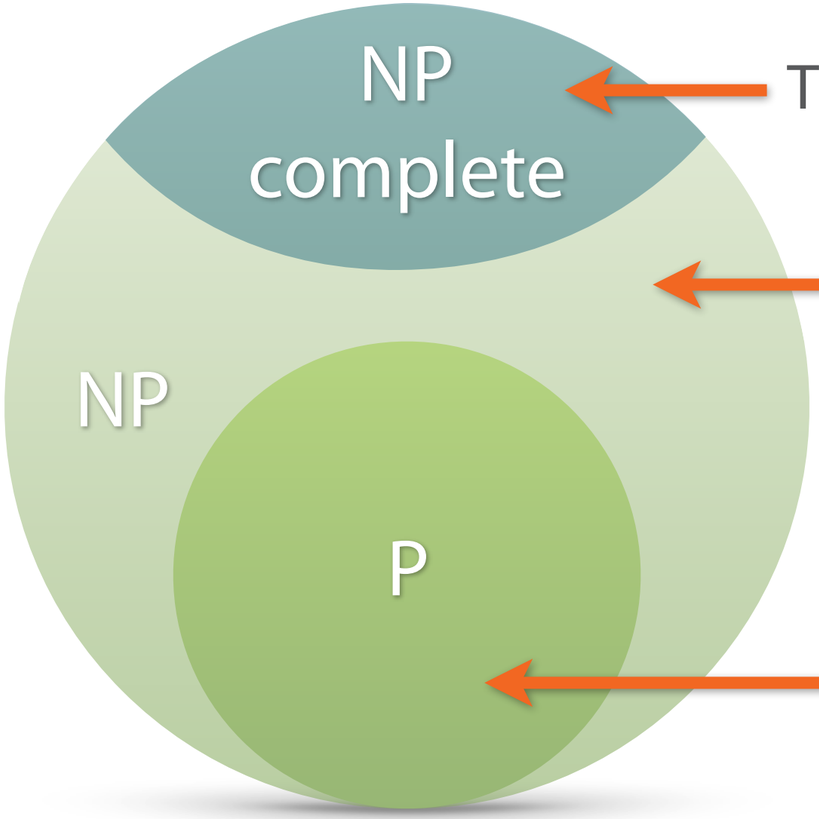
Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes → Indeed
No ↗

Certificate: a concrete route



The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Subset sum

Knapsack (can a value ≥*V* be achieved?)

Sorting

Finding shortest route

Traversing lists

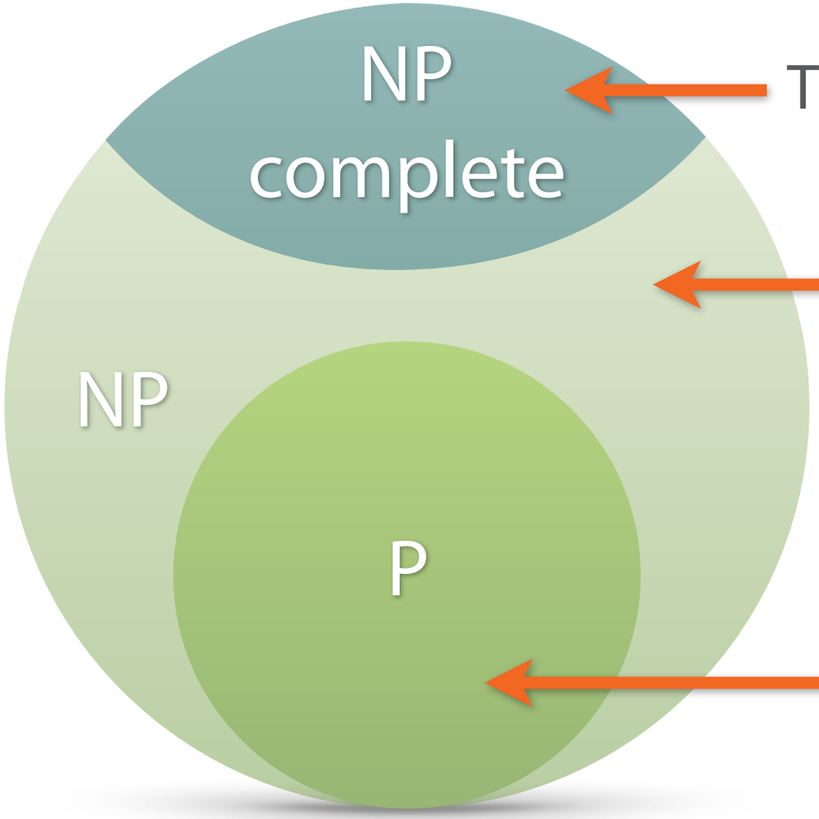
Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes → Indeed
No ↗

Certificate: a concrete route



The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Subset sum

Orthogonal packing problem

Knapsack (can a value ≥*V* be achieved?)

Sorting

Finding shortest route

Traversing lists

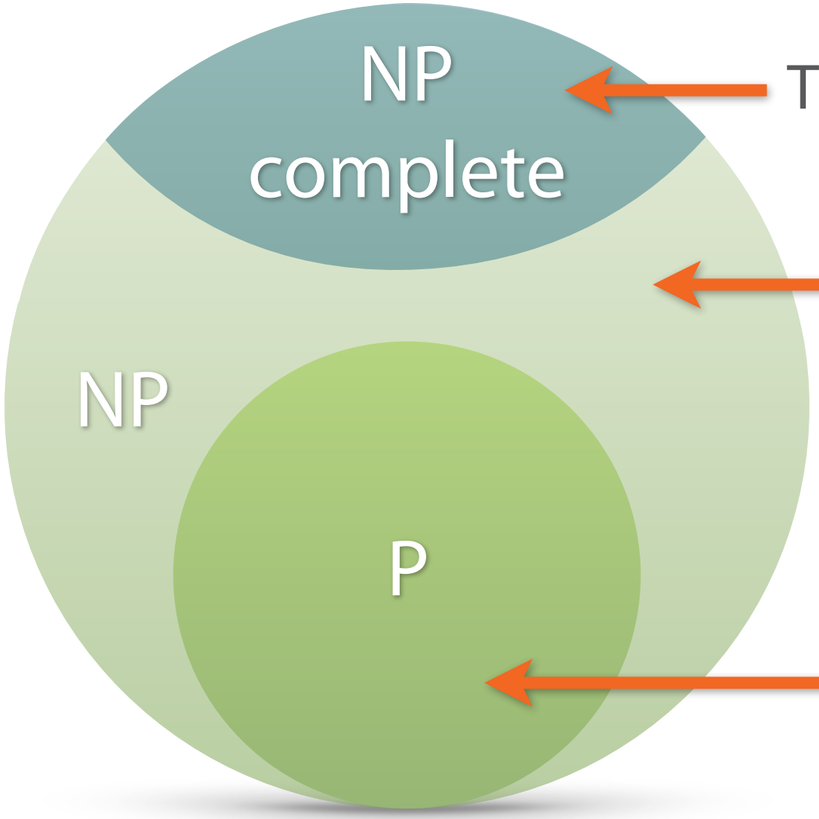
Look-ups in hash tables

P vs NP

Does a route shorter than *L* exist?

? Yes → Indeed
No ↗

Certificate: a concrete route



The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Subset sum

Orthogonal packing problem

Knapsack (can a value ≥*V* be achieved?)

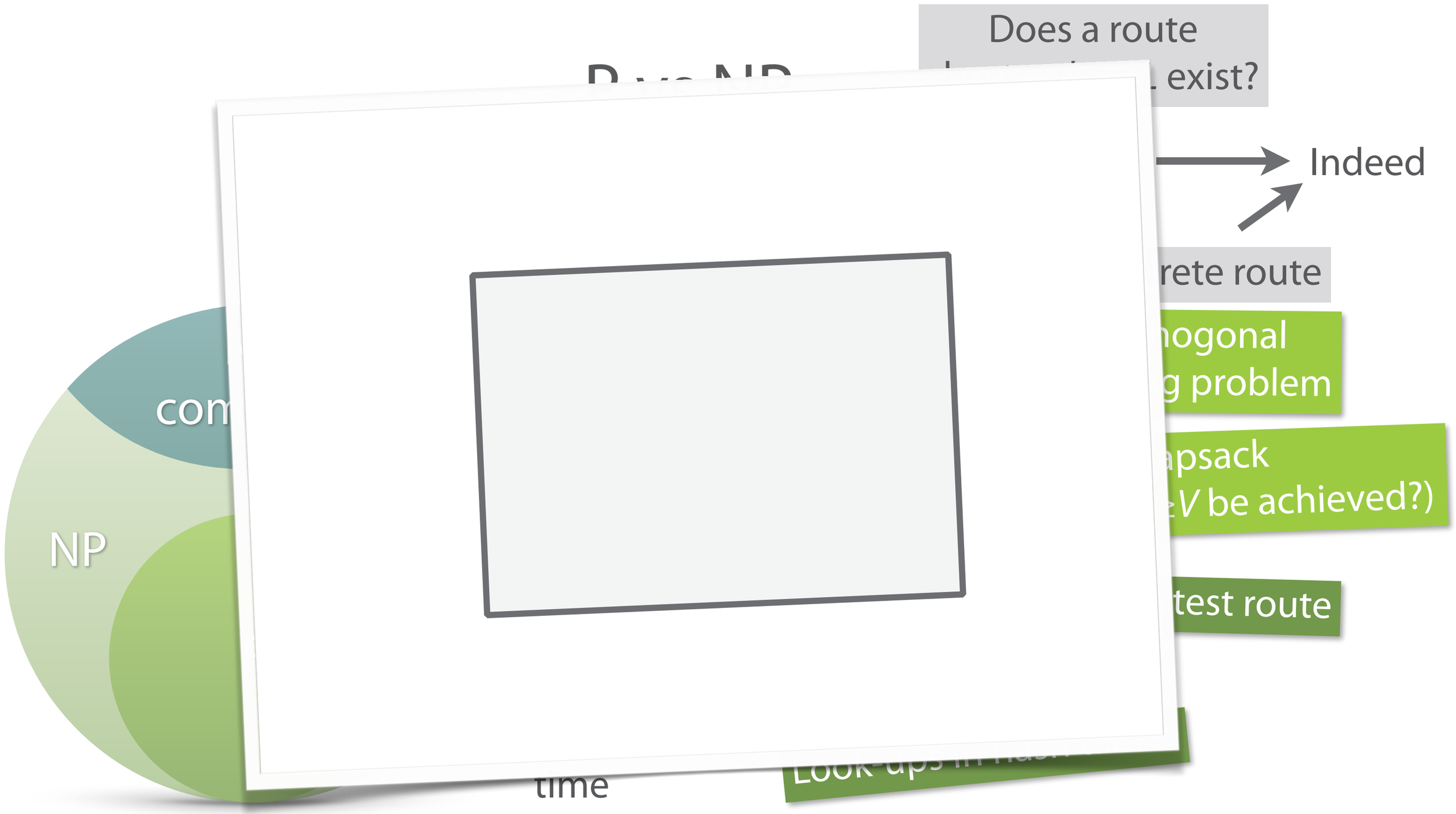
Sorting

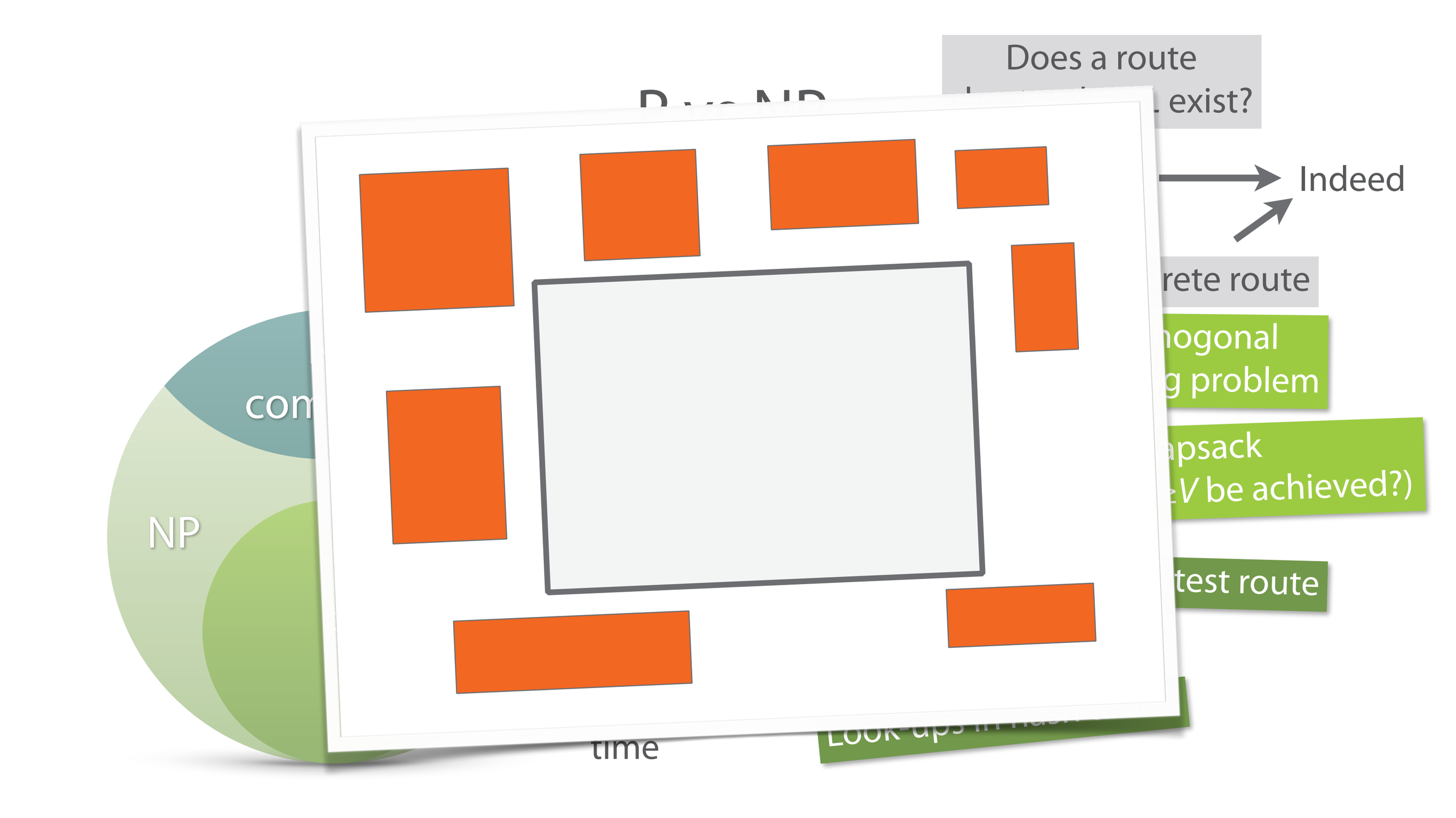
Finding shortest route

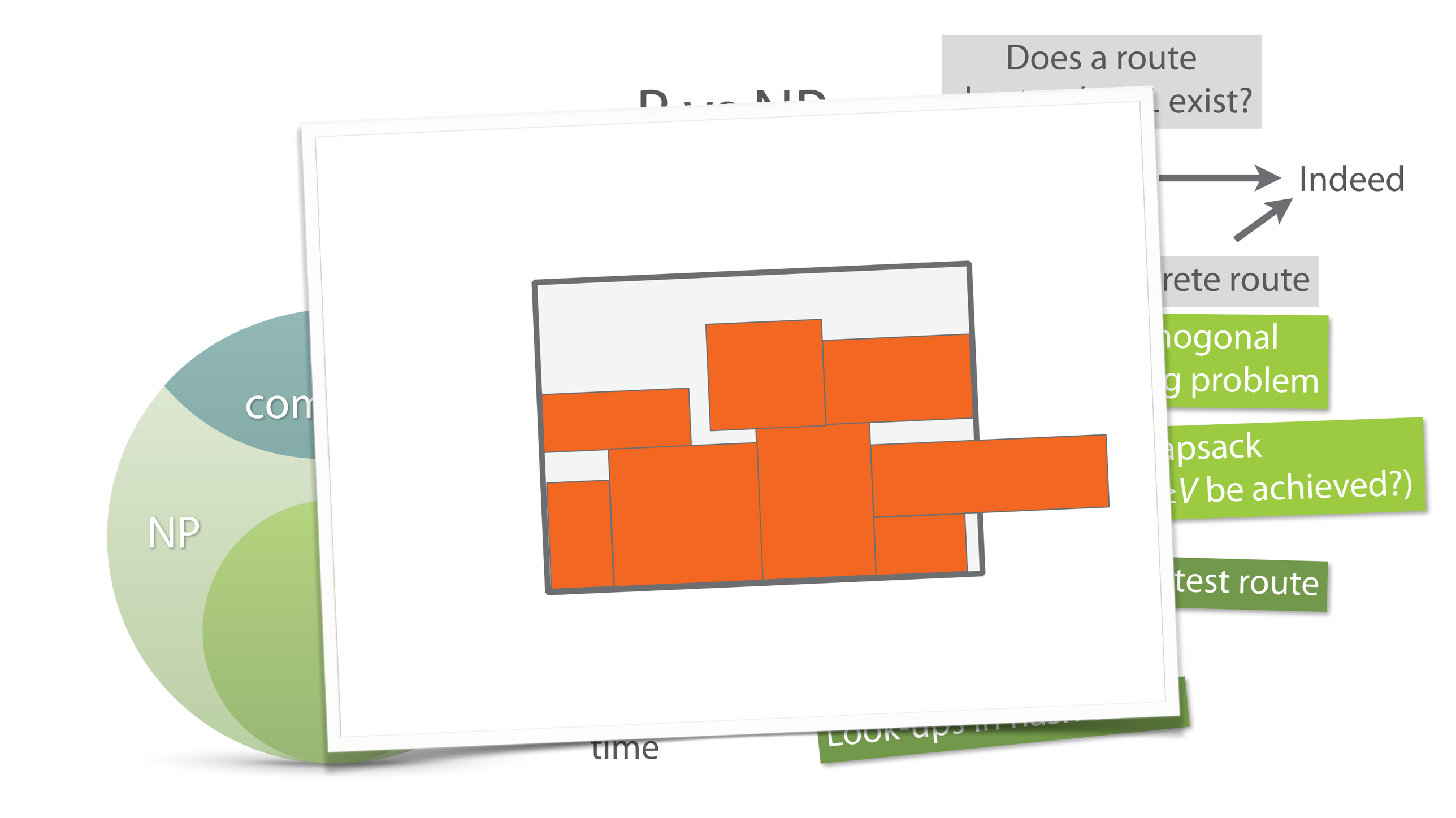
Traversing lists

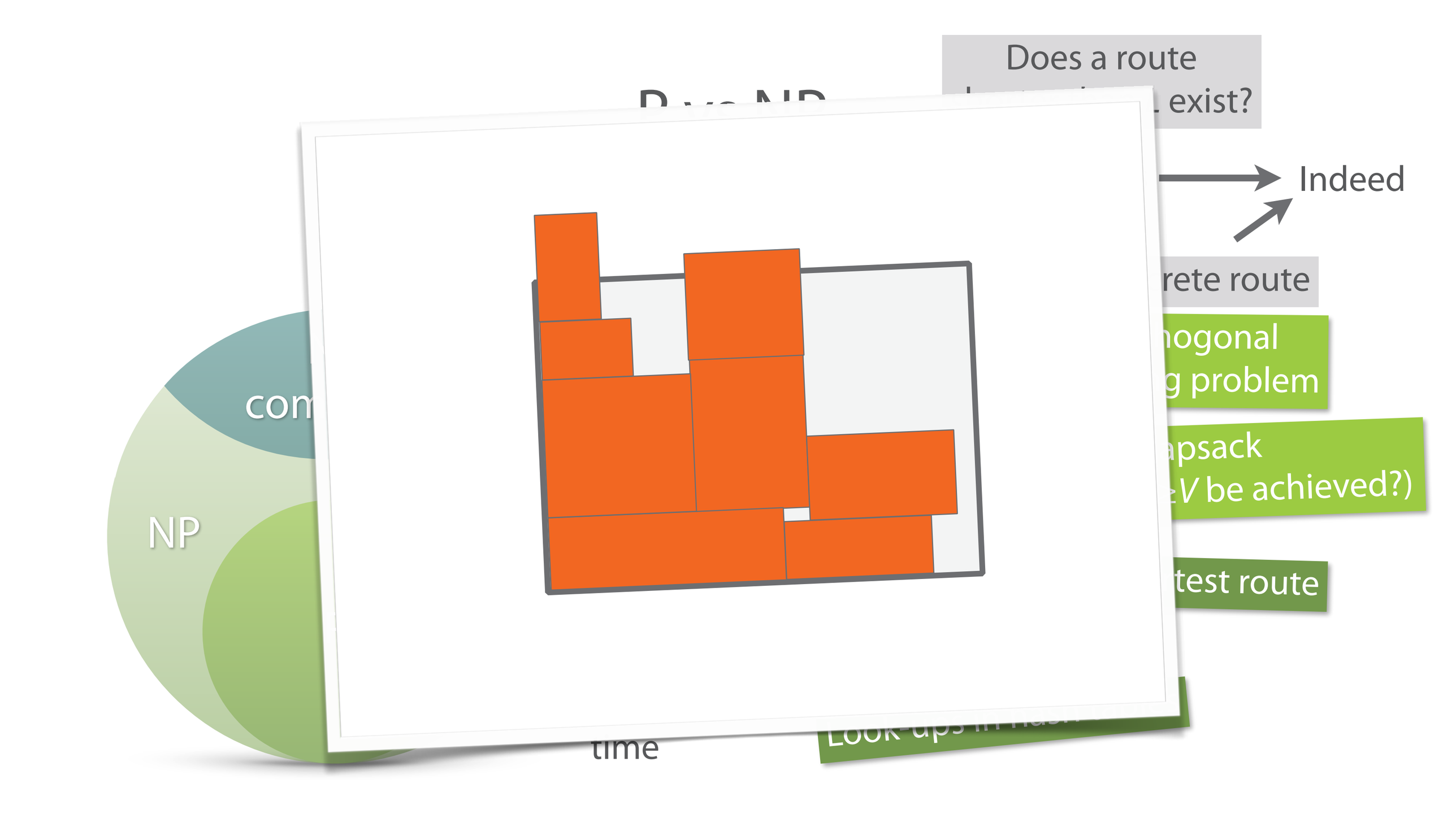
Look-ups in hash tables

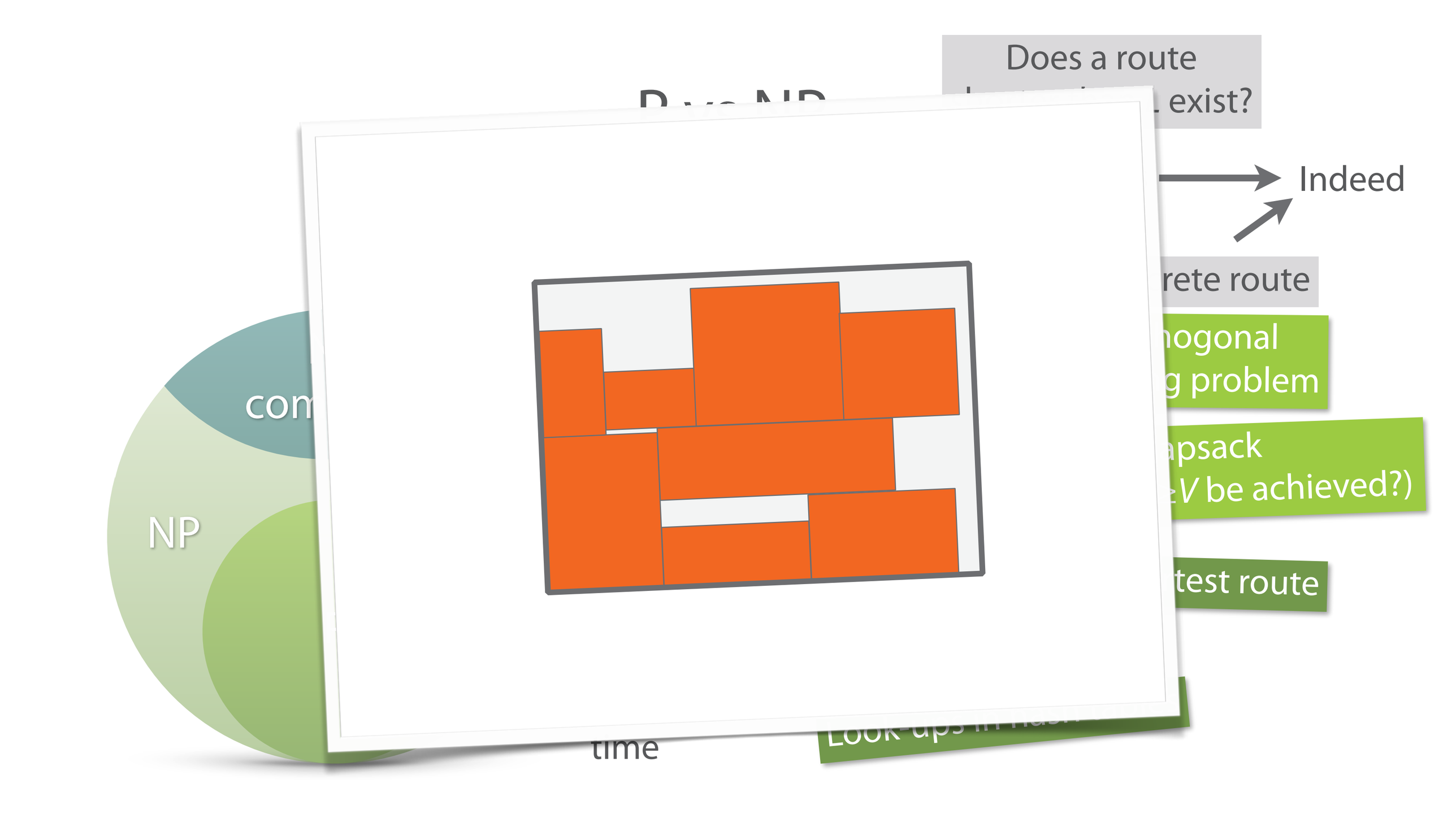

Does a route
exist?
Indeed
rete route
nogonal
g problem
apsack
≥V be achieved?)
test route
com
NP
time


Does a route
exist?
Indeed
rete route
nogonal
g problem
apsack
≥V be achieved?)
test route
com
NP
time


Does a route
exist?
Indeed
rete route
nogonal
g problem
apsack
≥V be achieved?)
test route
com
NP
time


Does a route
exist?
Indeed
rete route
hogonal
g problem
apsack
≥V be achieved?)
test route
com
NP
time


Does a route
exist?
Indeed
rete route
nogonal
g problem
apsack
≥V be achieved?)
test route
com
NP
time

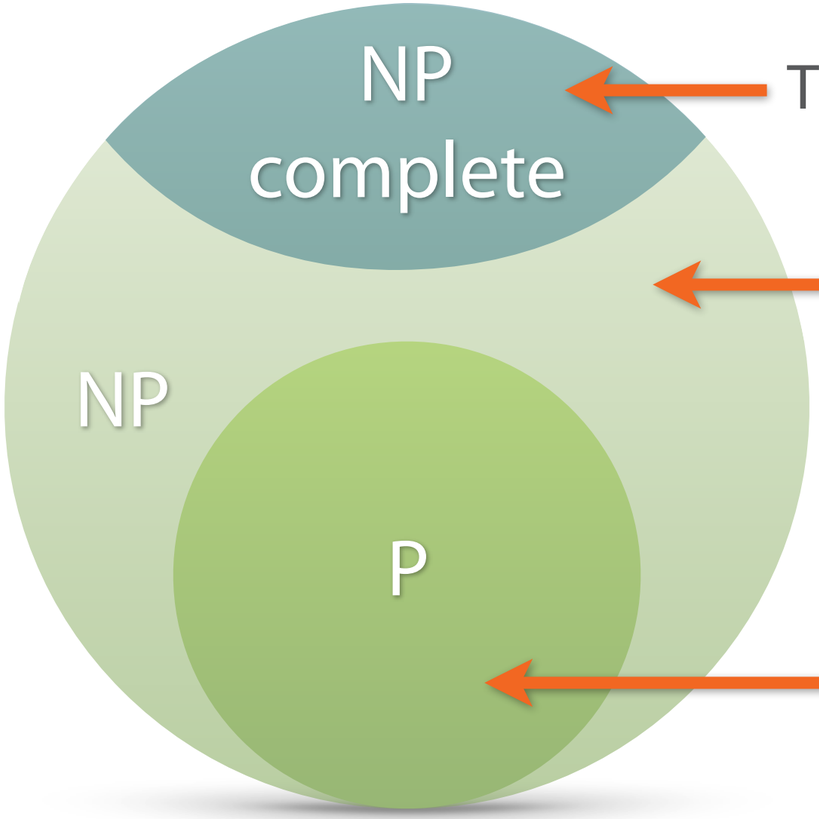

Does a route
exist?
Indeed
rete route
nogonal
g problem
apsack
≥V be achieved?)
test route
com
NP
time

P vs NP

Does a route shorter than *L* exist?

? Yes → Indeed
No ↗

Certificate: a concrete route



The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Subset sum

Orthogonal packing problem

Knapsack (can a value ≥*V* be achieved?)

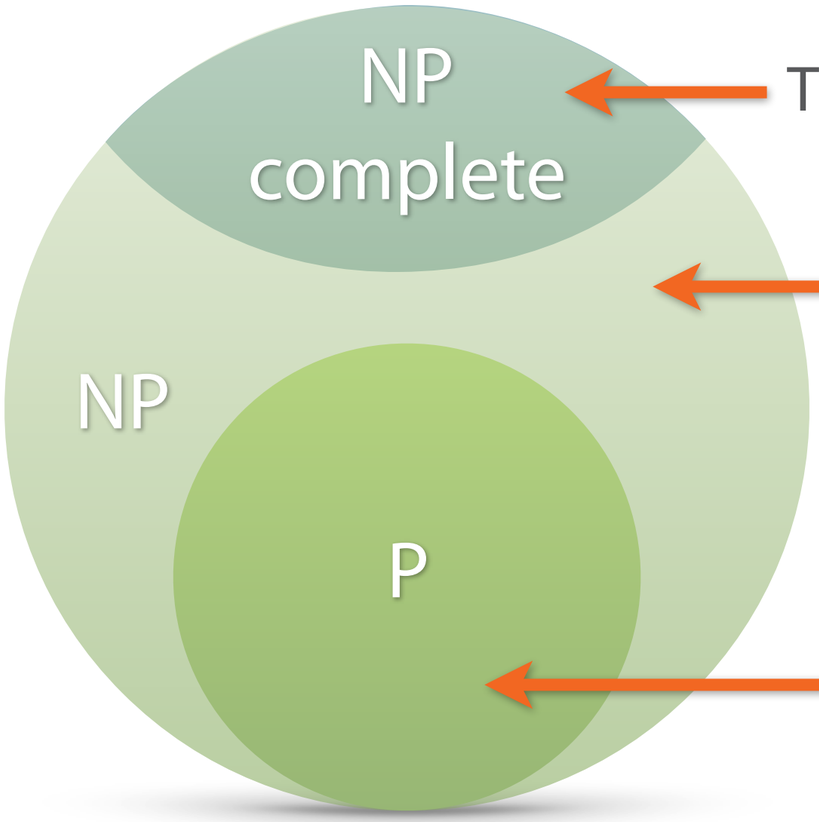
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Subset sum

Orthogonal packing problem

Knapsack (decision)
(can a value ≥*V* be achieved?)

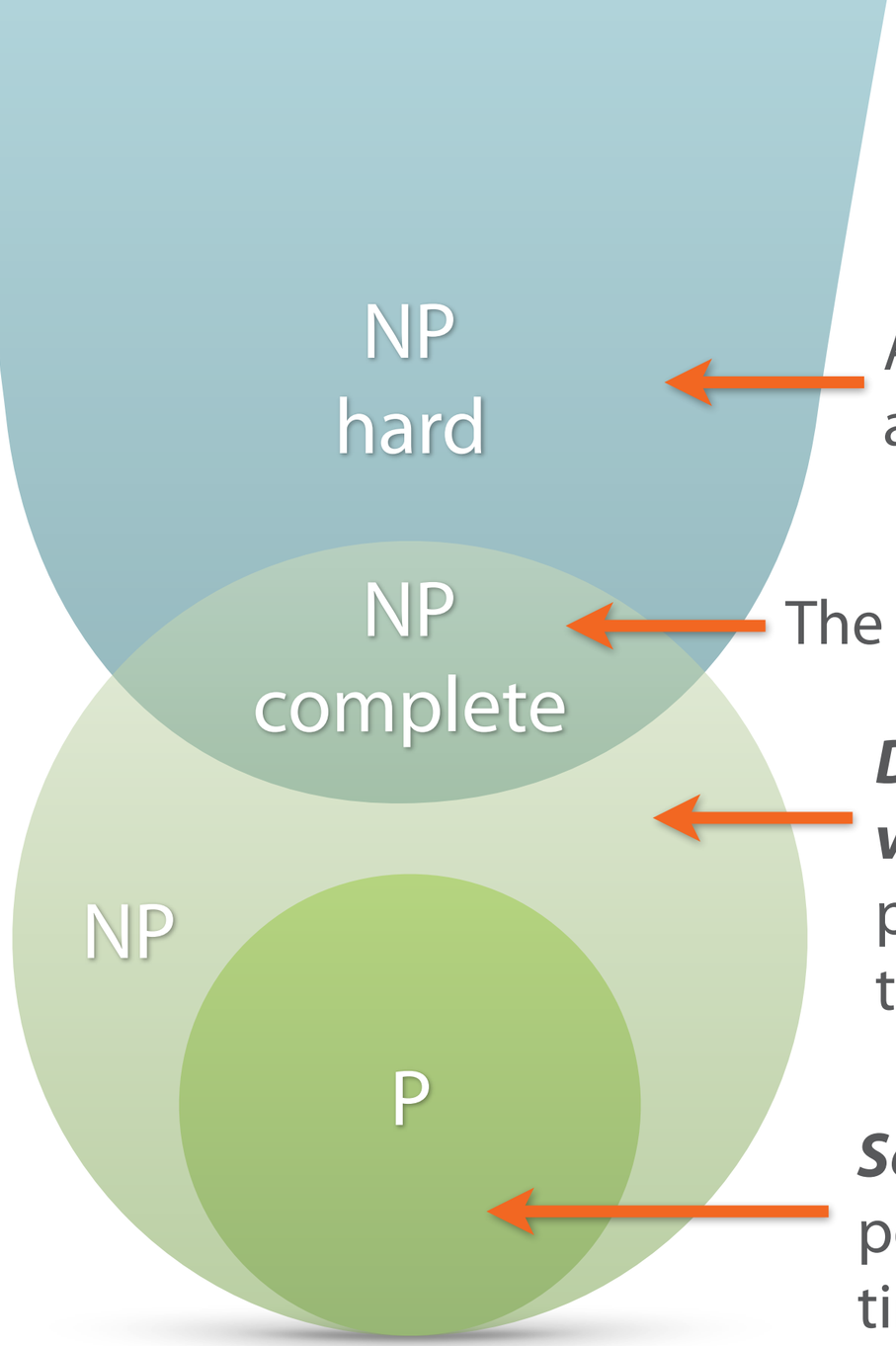
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



At least as hard as NP complete

The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Subset sum

Orthogonal packing problem

Knapsack (decision) (can a value ≥*V* be achieved?)

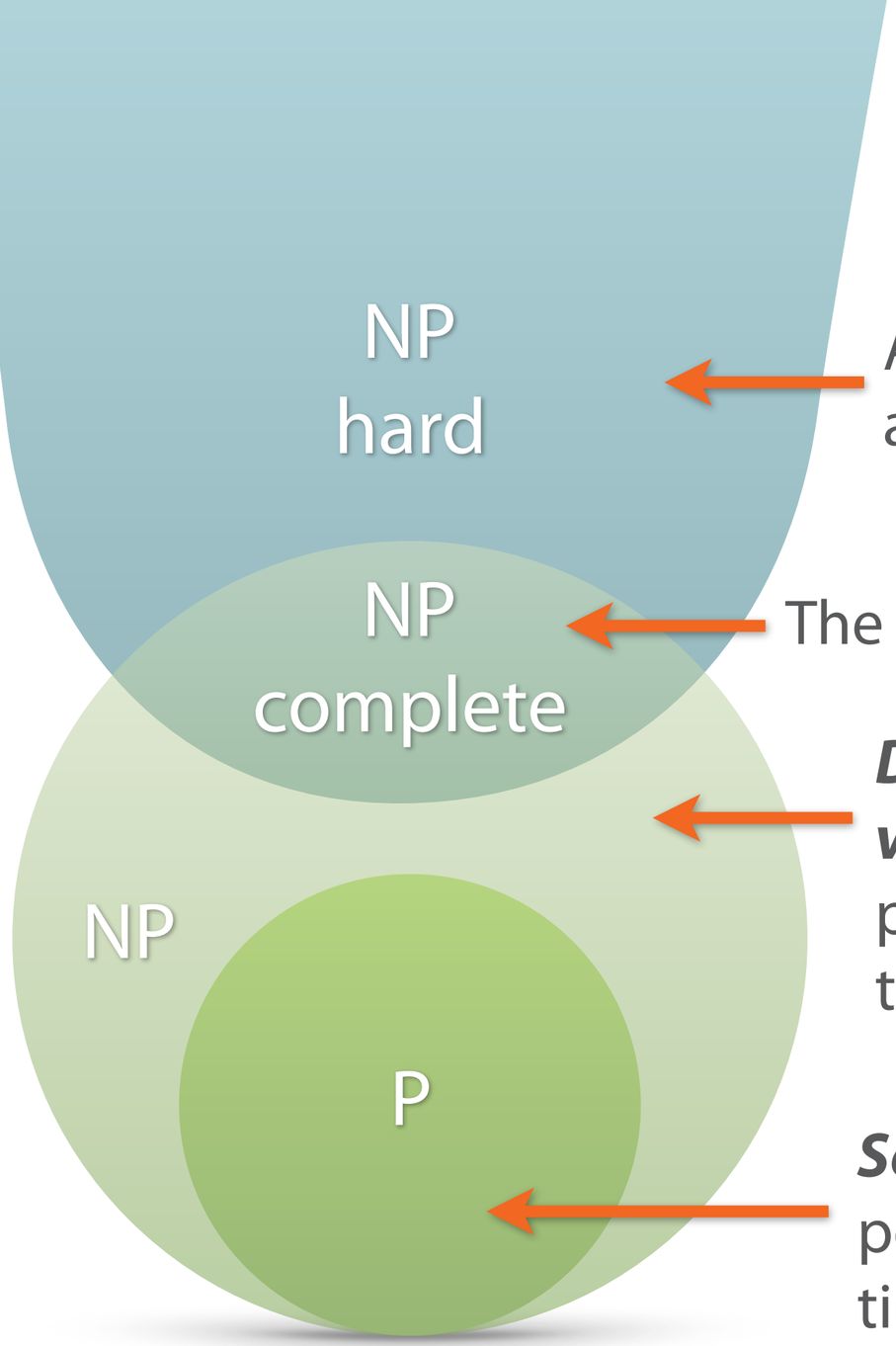
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



At least as hard as NP complete

The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Knapsack (optimization) (maximize *V*)

Subset sum

Orthogonal packing problem

Knapsack (decision) (can a value ≥*V* be achieved?)

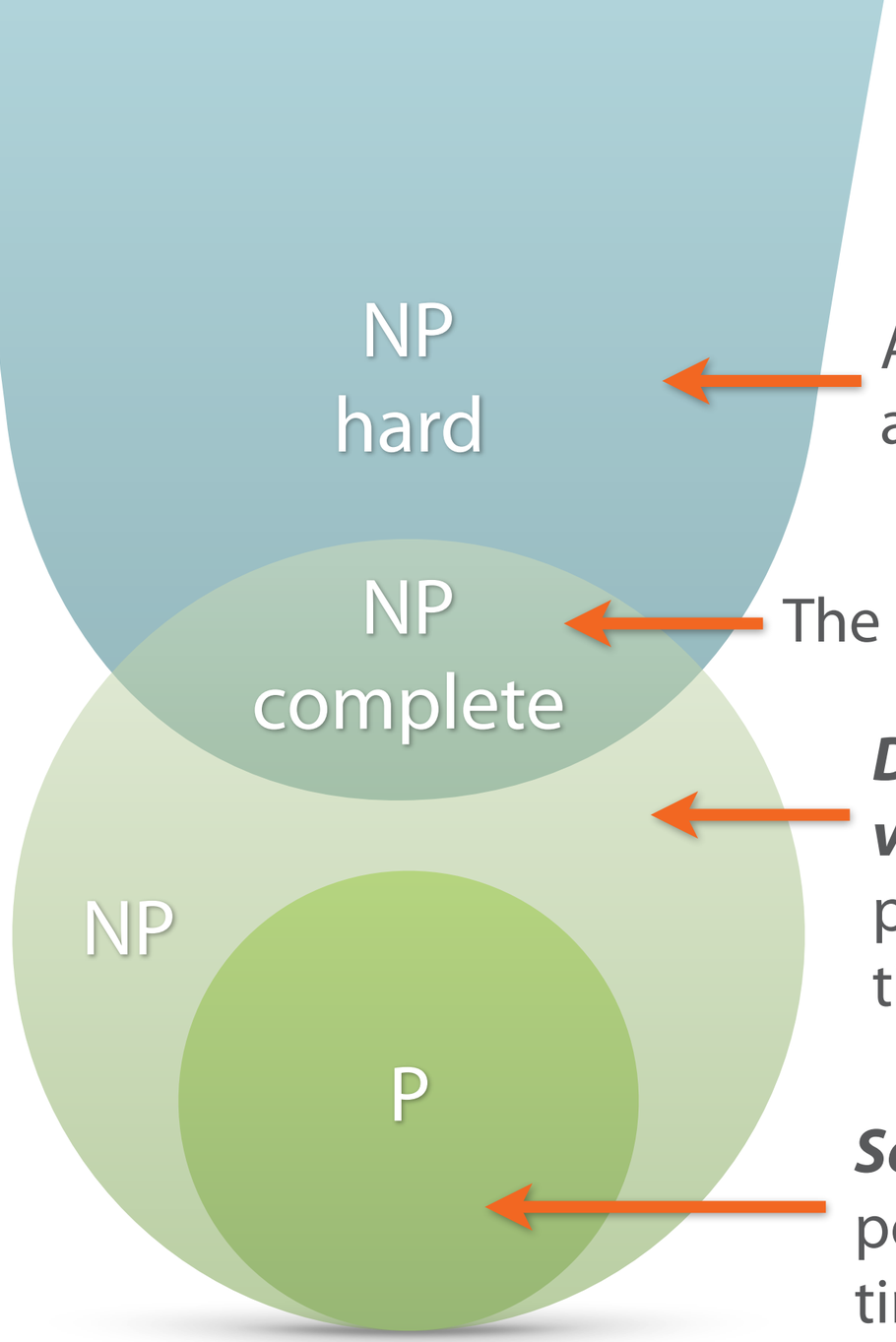
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



At least as hard as NP complete

The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Knapsack (optimization) (maximize *V*)

Halting problem

Subset sum

Orthogonal packing problem

Knapsack (decision) (can a value ≥*V* be achieved?)

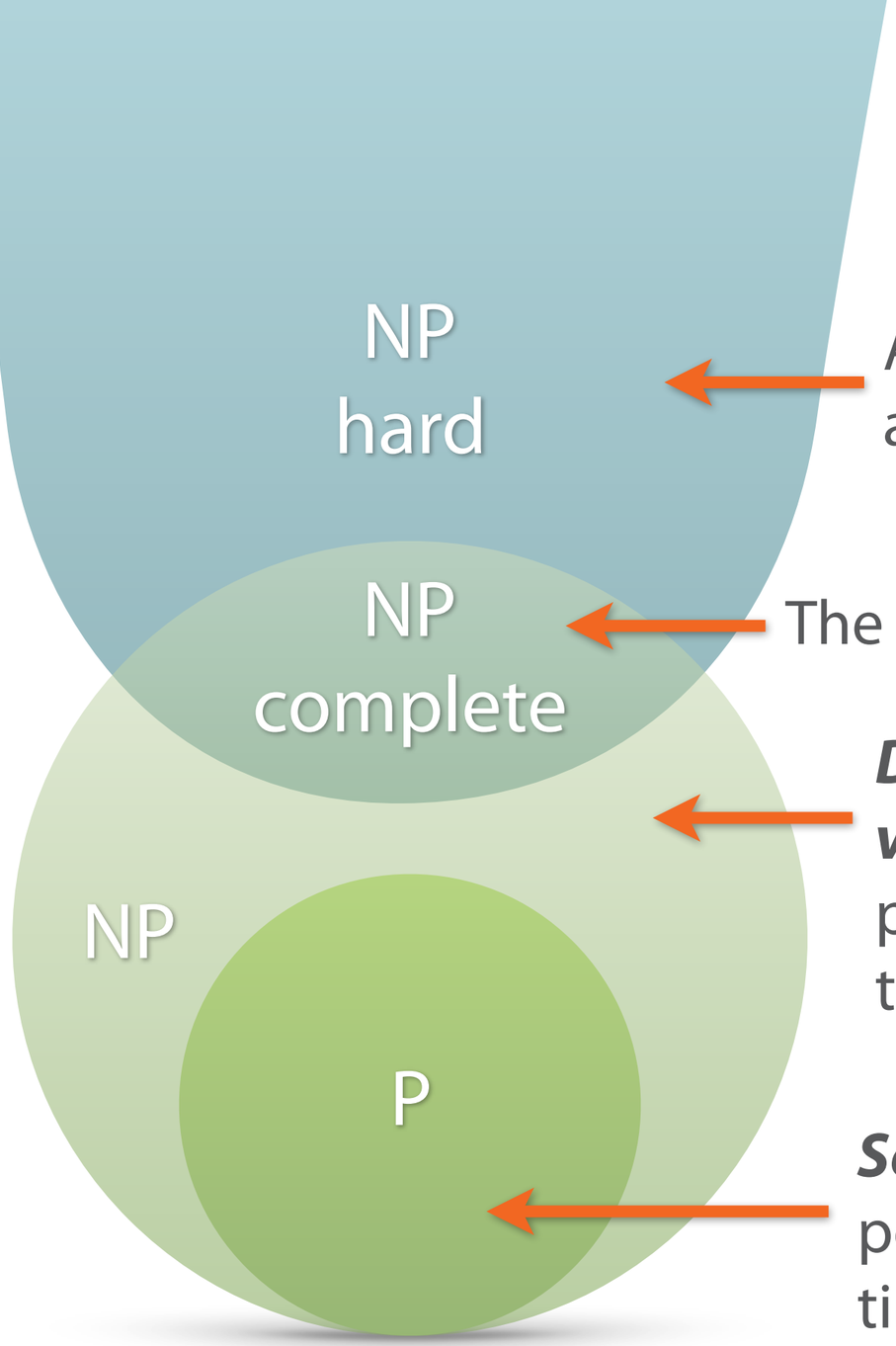
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



At least as hard as NP complete

Knapsack (optimization) (maximize *V*)

Halting problem

Traveling salesman

The hardest in NP

Subset sum

Orthogonal packing problem

Decision problems ***verifiable*** in polynomial time

Knapsack (decision) (can a value ≥*V* be achieved?)

Sorting

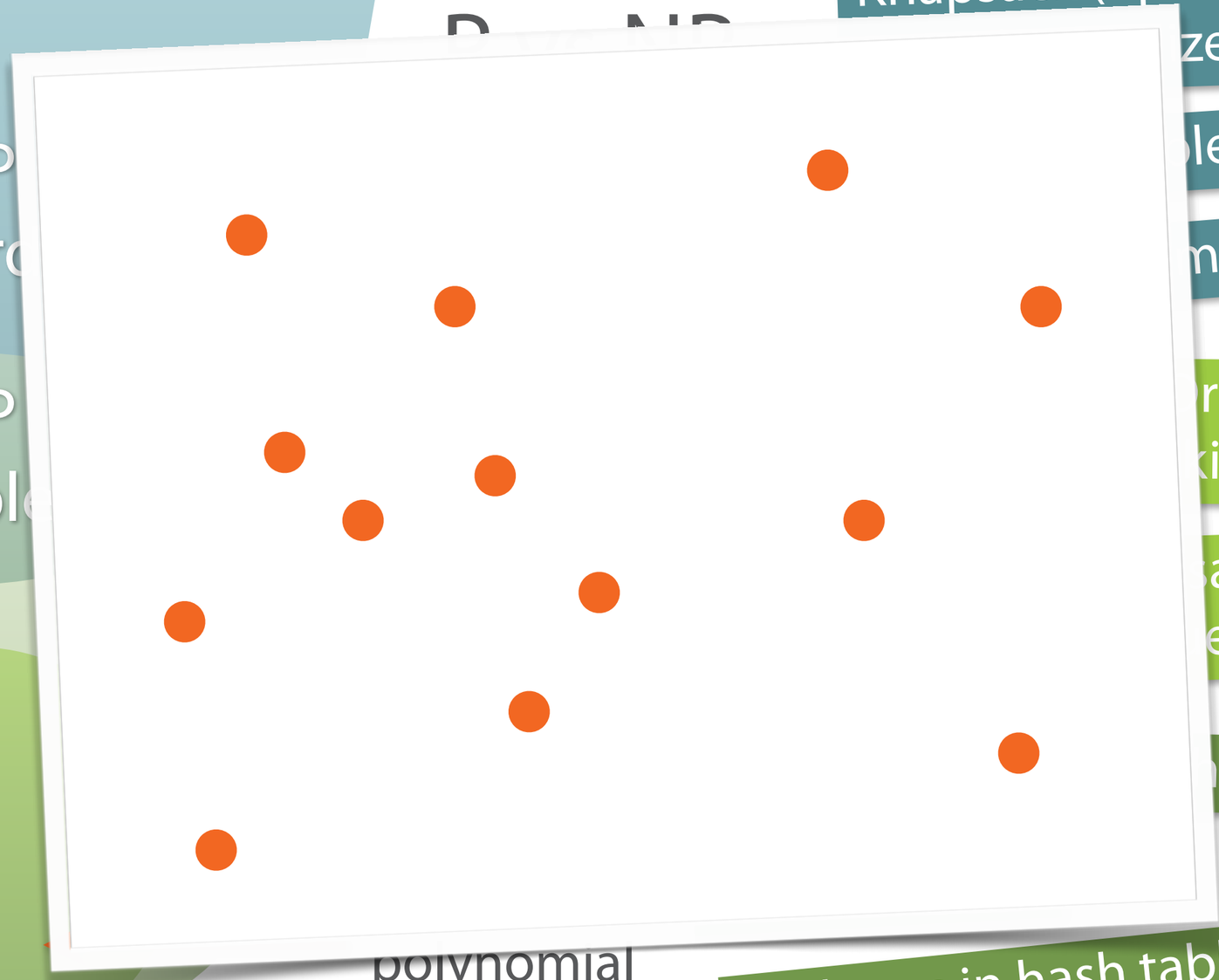
Finding shortest route

Solvable in polynomial time

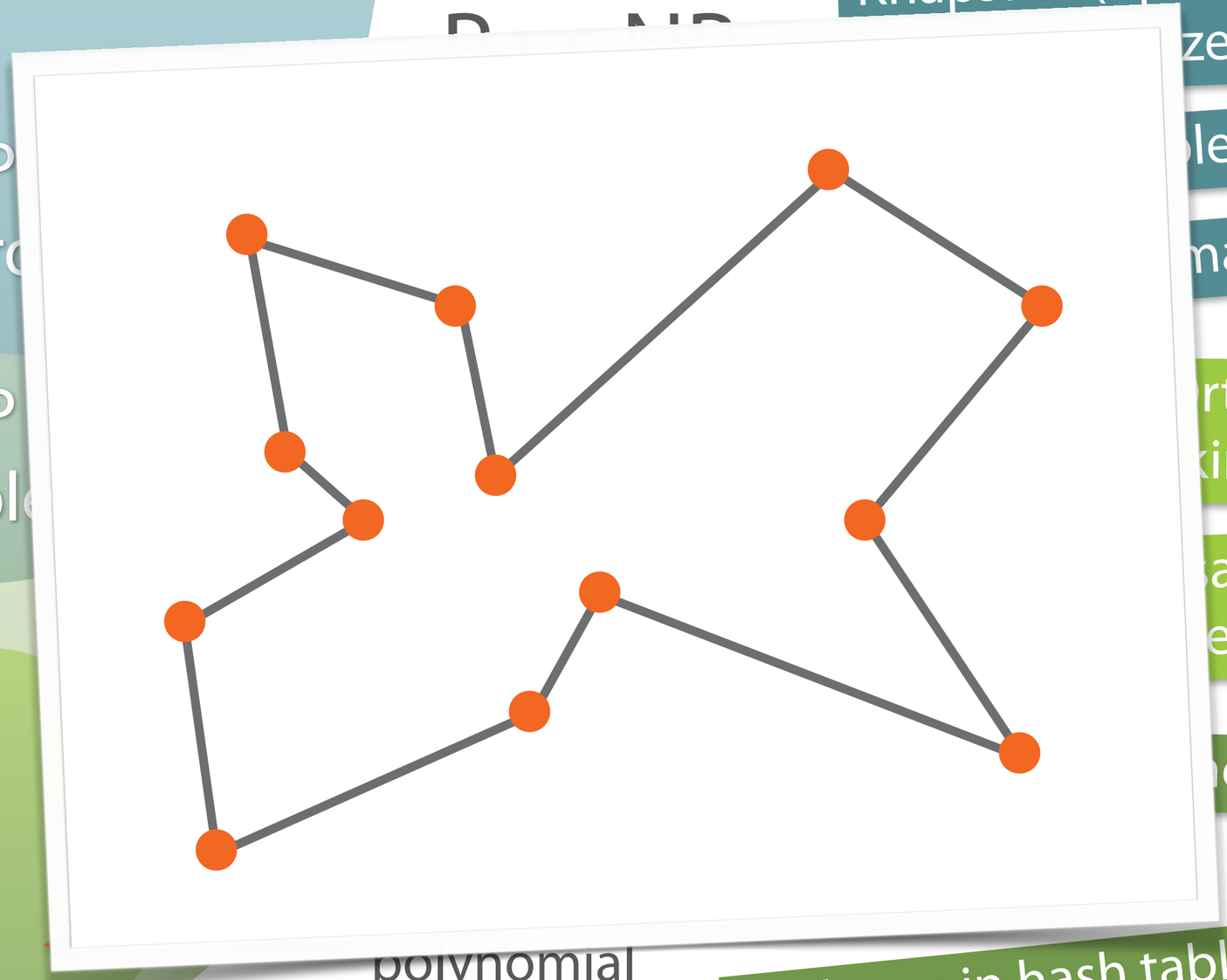
Traversing lists

Look-ups in hash tables

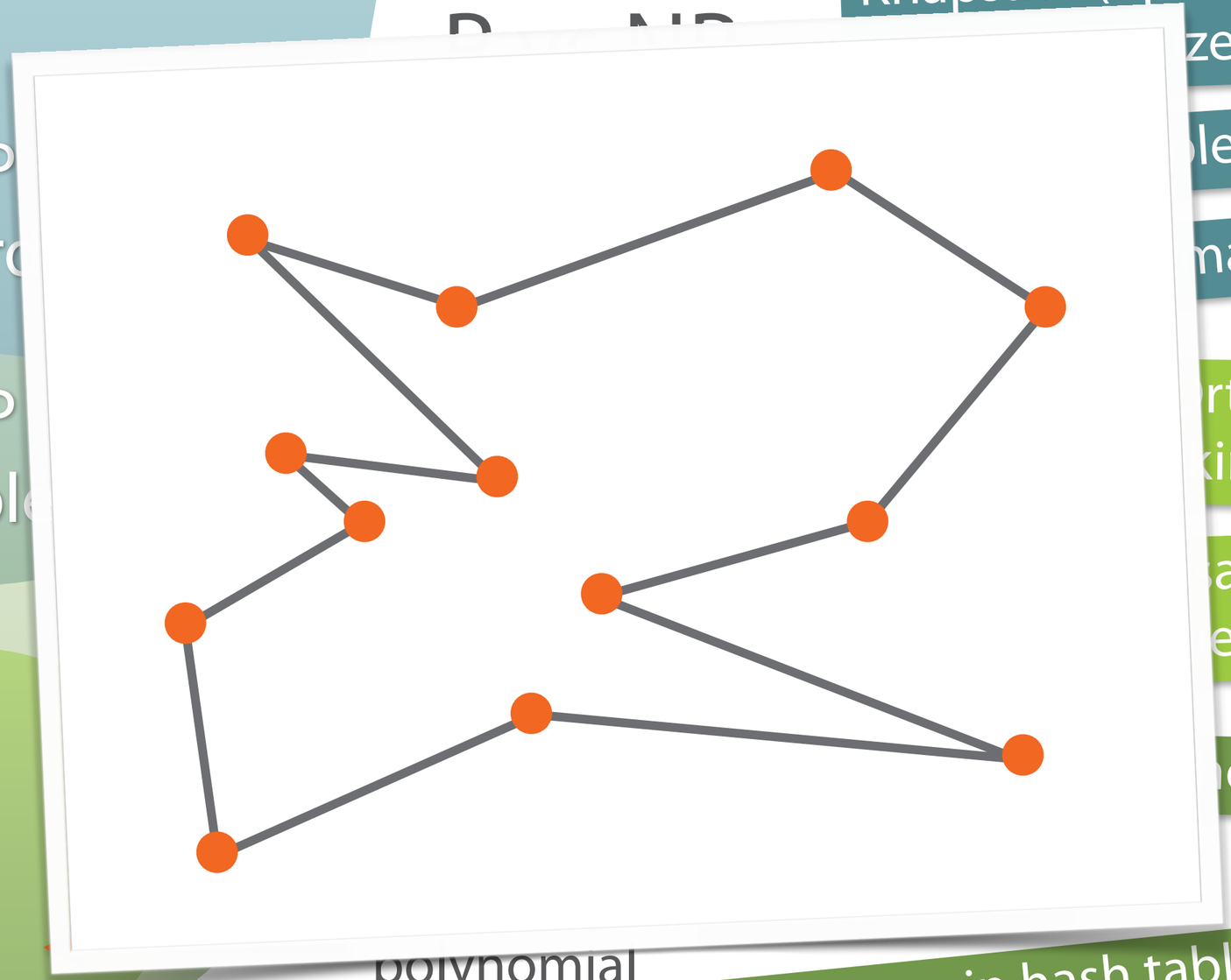

NP
harc
NP
comple
NP
P
polynomial
time
Knapsack (optimization)
ze V)
lem
man
rthogonal
king problem
ack (decision)
e ≥V be achieved?)
ortest route
Look-ups in hash tables



Knapsack (optimization)
ze *V*)
lem
man
rthogonal
king problem
ack (decision)
e ≥*V* be achieved?)
ortest route
Look-ups in hash tables
NP
harc
NP
comple
NP
P
polynomial
time



Knapsack (optimization)
ze *V*)
lem
man
rthogonal
king problem
ack (decision)
e ≥*V* be achieved?)
ortest route
Look-ups in hash tables
NP
harc
NP
comple
NP
P
polynomial
time



Knapsack (optimization)
ze *V*)

lem

man

rthogonal
king problem

ack (decision)
e ≥*V* be achieved?)

ortest route

Look-ups in hash tables

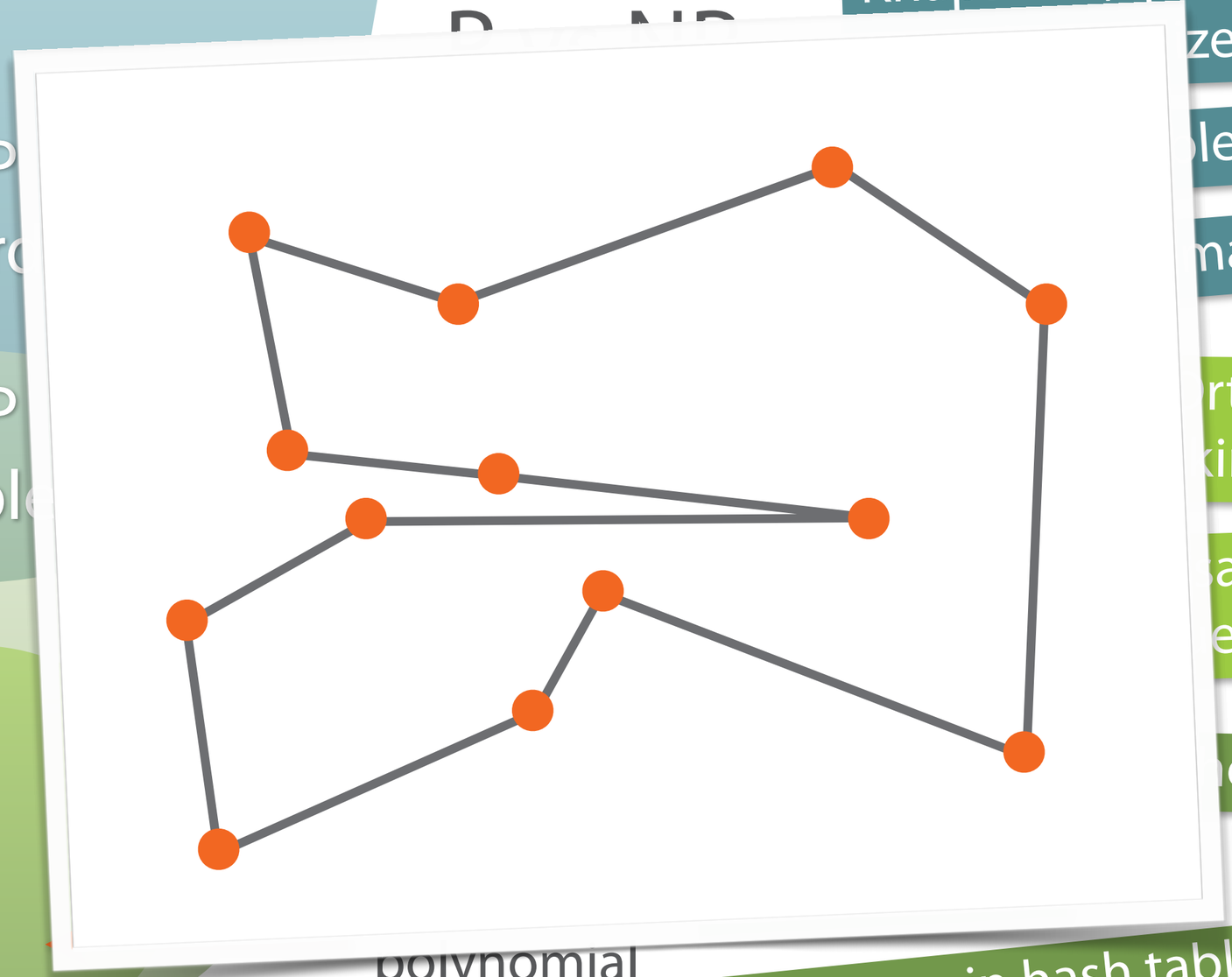
NP
harc

NP
comple

NP

P

polynomial
time



Knapsack (optimization)
ze *V*)

lem

man

rthogonal
king problem

ack (decision)
e ≥*V* be achieved?)

ortest route

Look-ups in hash tables

NP
harc

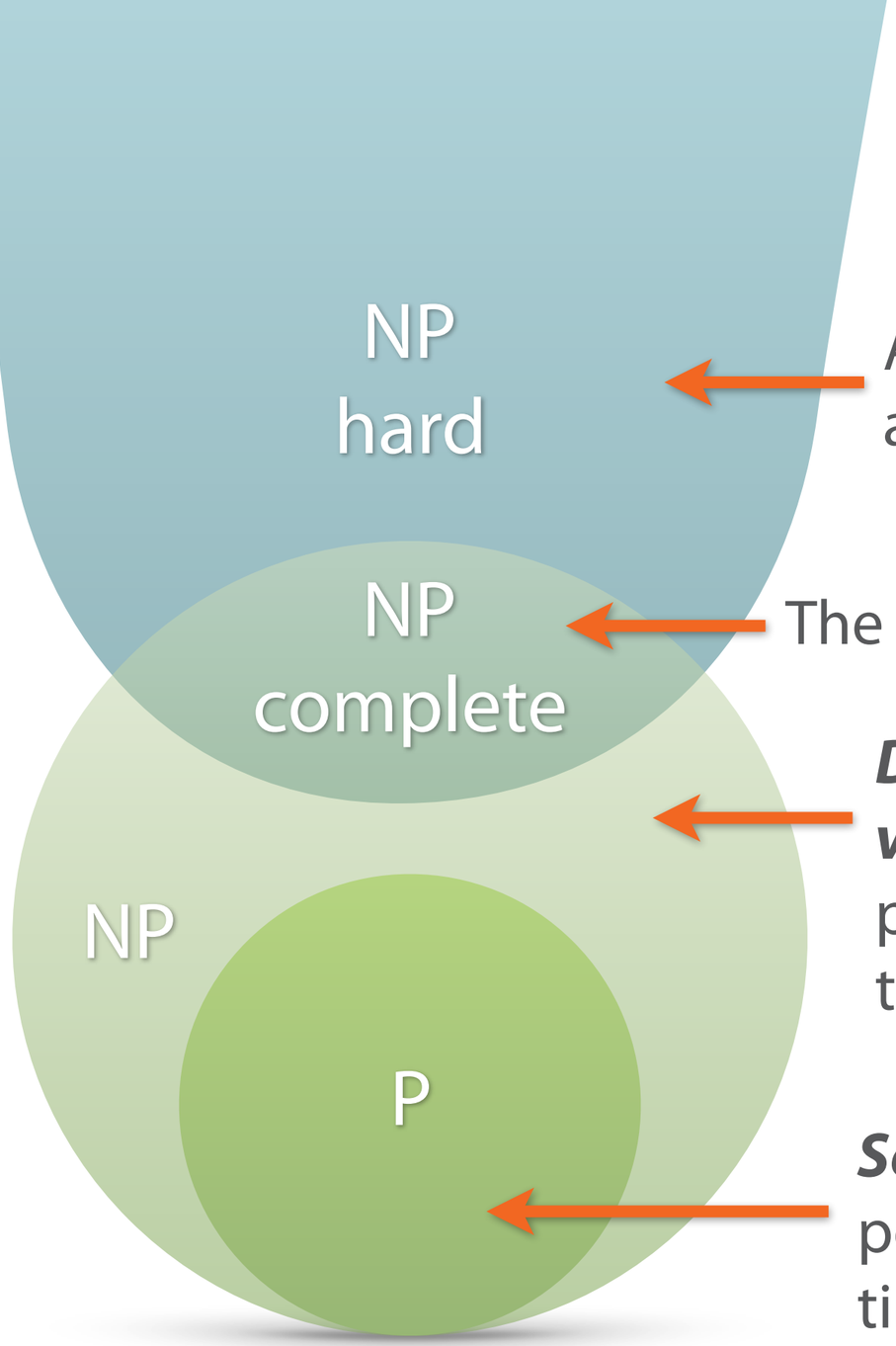
NP
comple

NP

P

polynomial
time

P vs NP



At least as hard as NP complete

Knapsack (optimization) (maximize *V*)

Halting problem

Traveling salesman

The hardest in NP

Orthogonal packing problem

Subset sum

Decision problems ***verifiable*** in polynomial time

Knapsack (decision) (can a value ≥*V* be achieved?)

Sorting

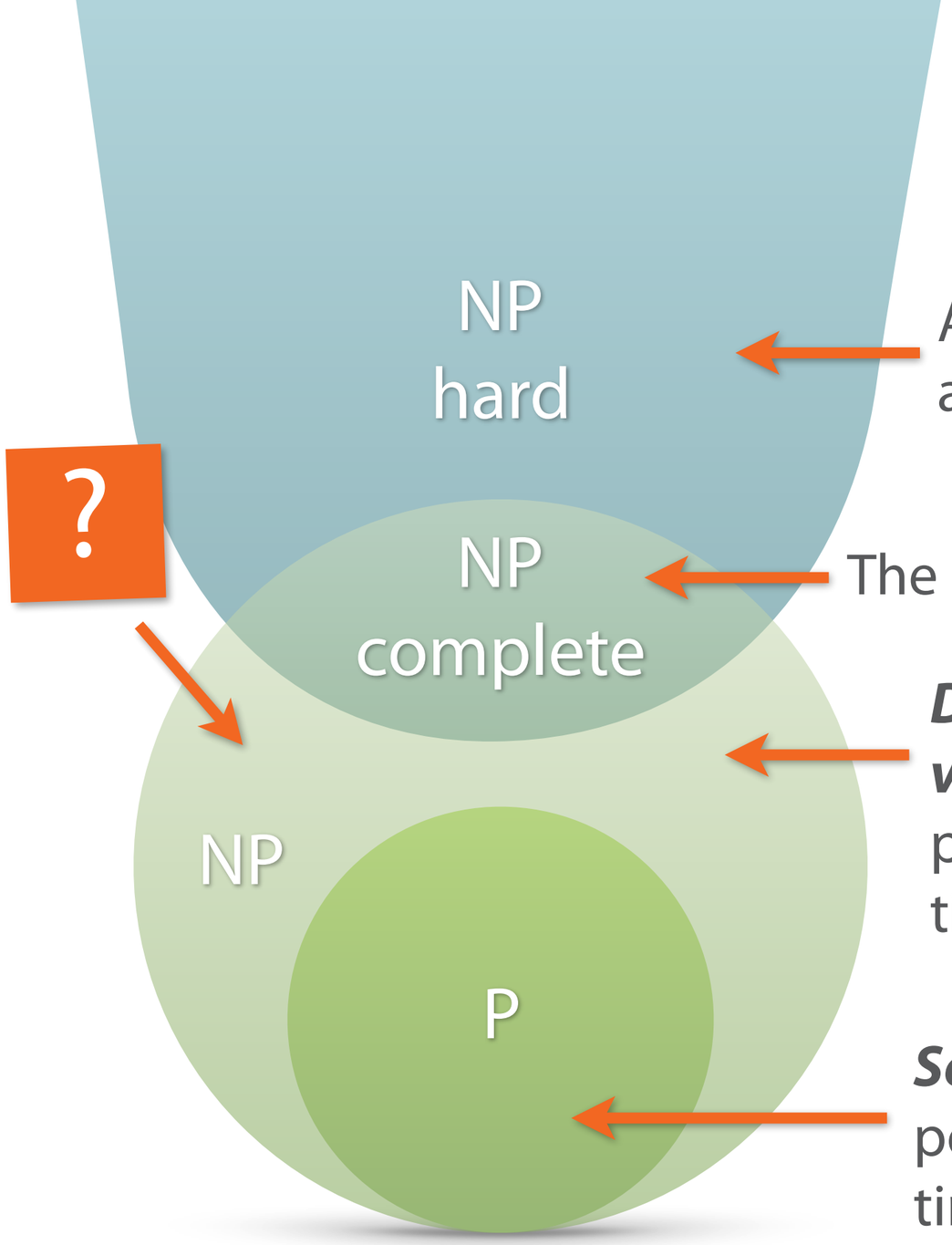
Finding shortest route

Solvable in polynomial time

Traversing lists

Look-ups in hash tables

P vs NP



At least as hard as NP complete

The hardest in NP

Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

Knapsack (optimization) (maximize *V*)

Halting problem

Traveling salesman

Subset sum

Orthogonal packing problem

Knapsack (decision) (can a value ≥*V* be achieved?)

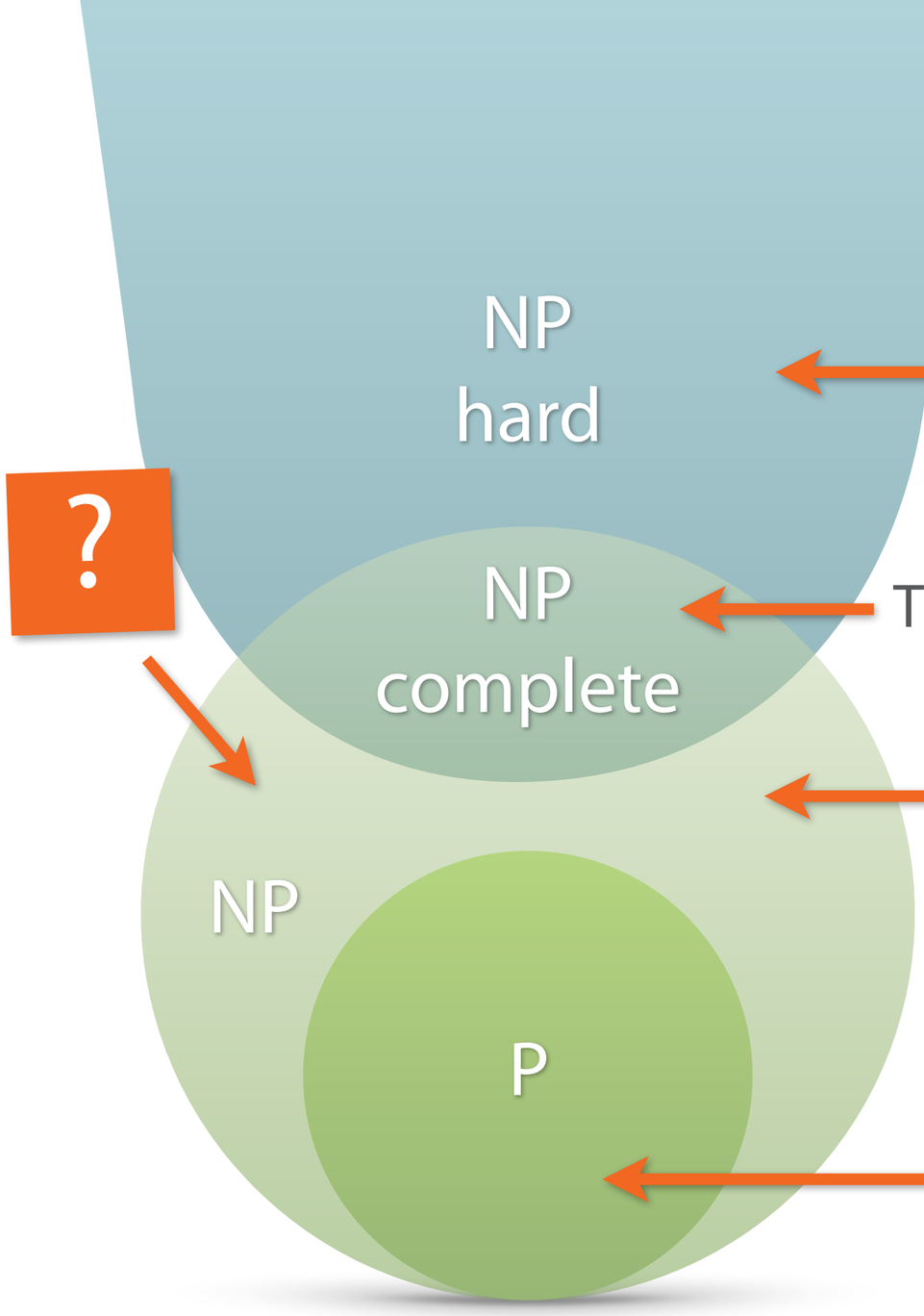
Sorting

Finding shortest route

Traversing lists

Look-ups in hash tables

P vs NP



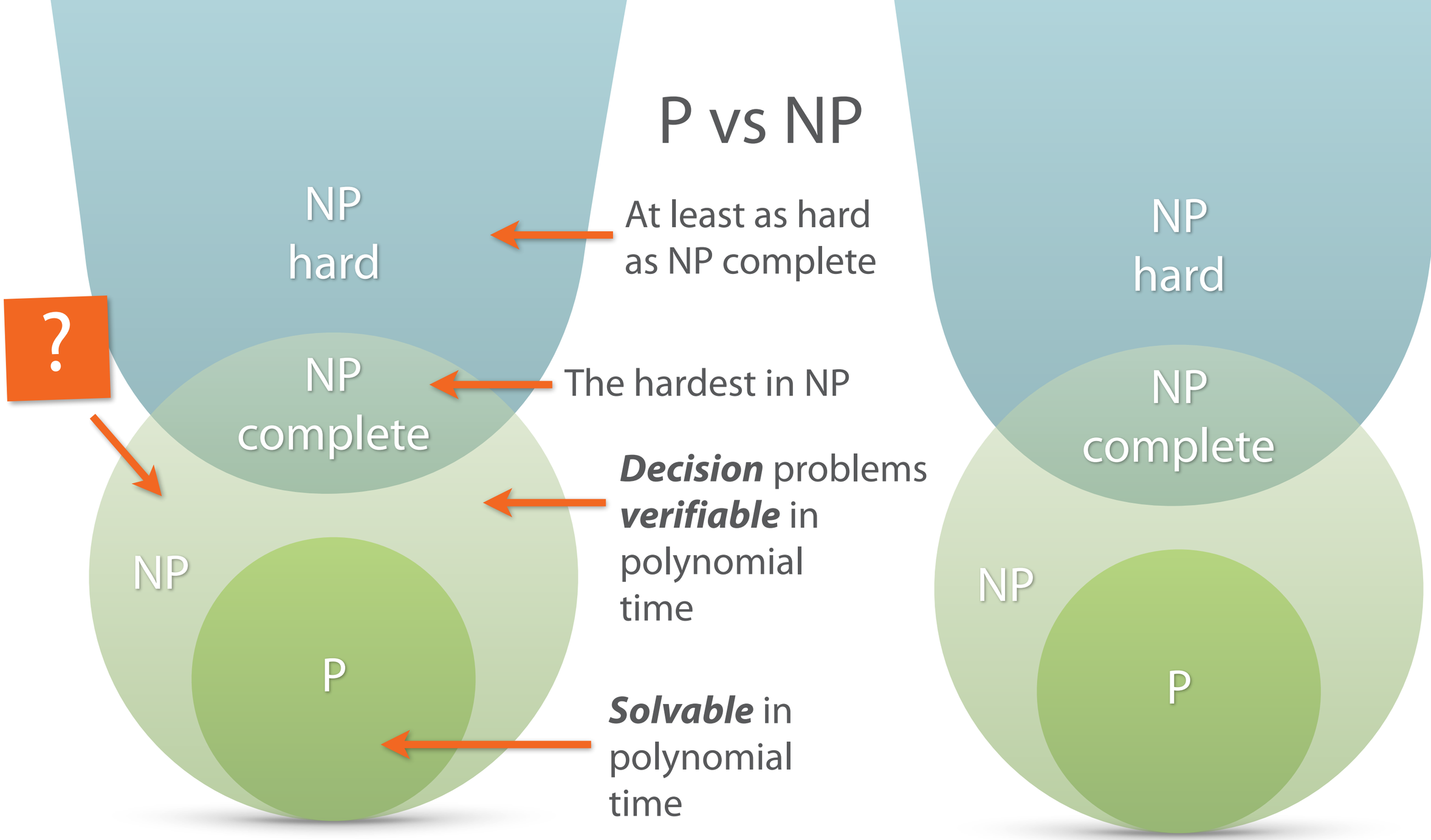
At least as hard as NP complete

The hardest in NP

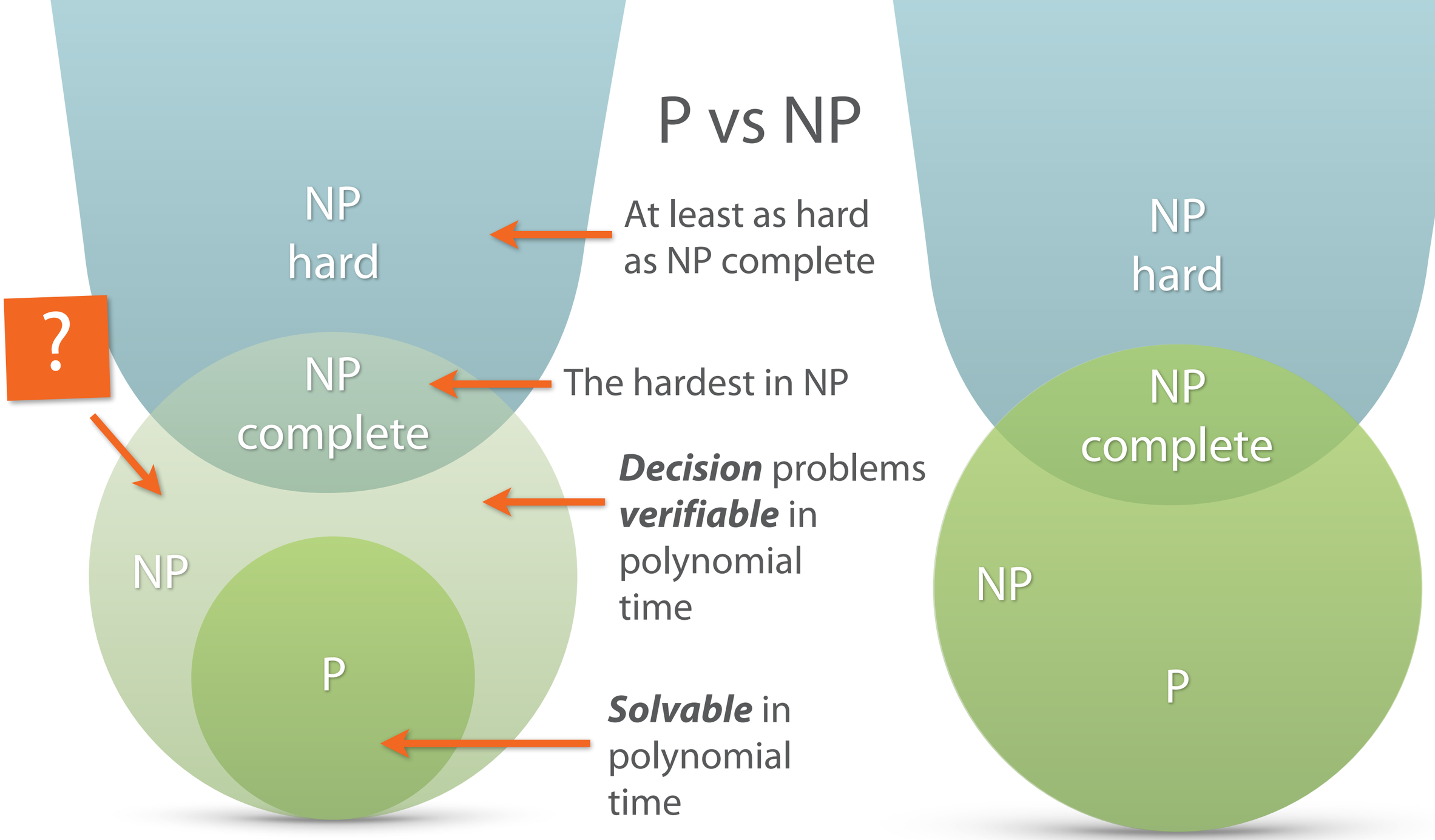
Decision problems ***verifiable*** in polynomial time

Solvable in polynomial time

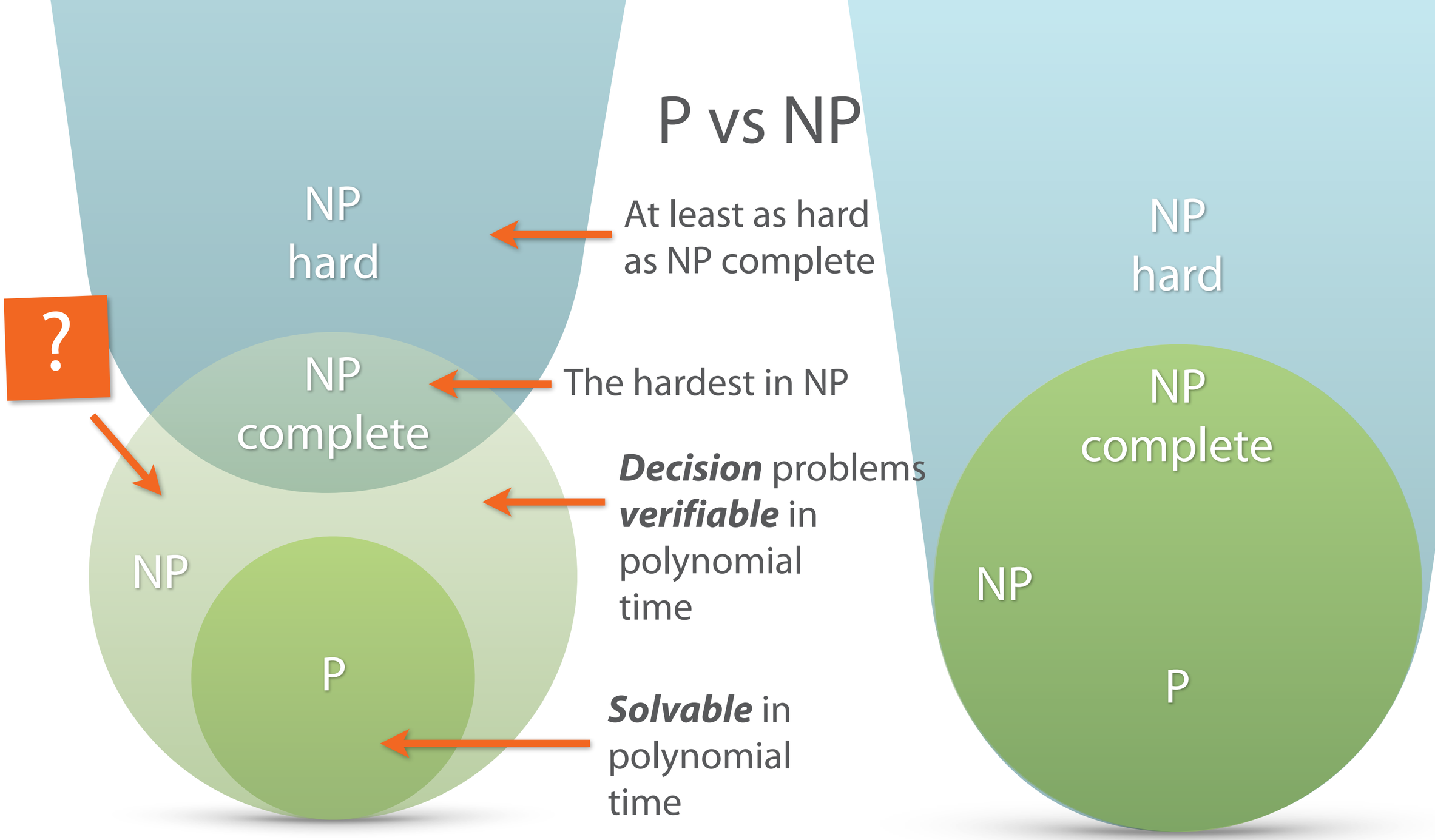
P vs NP

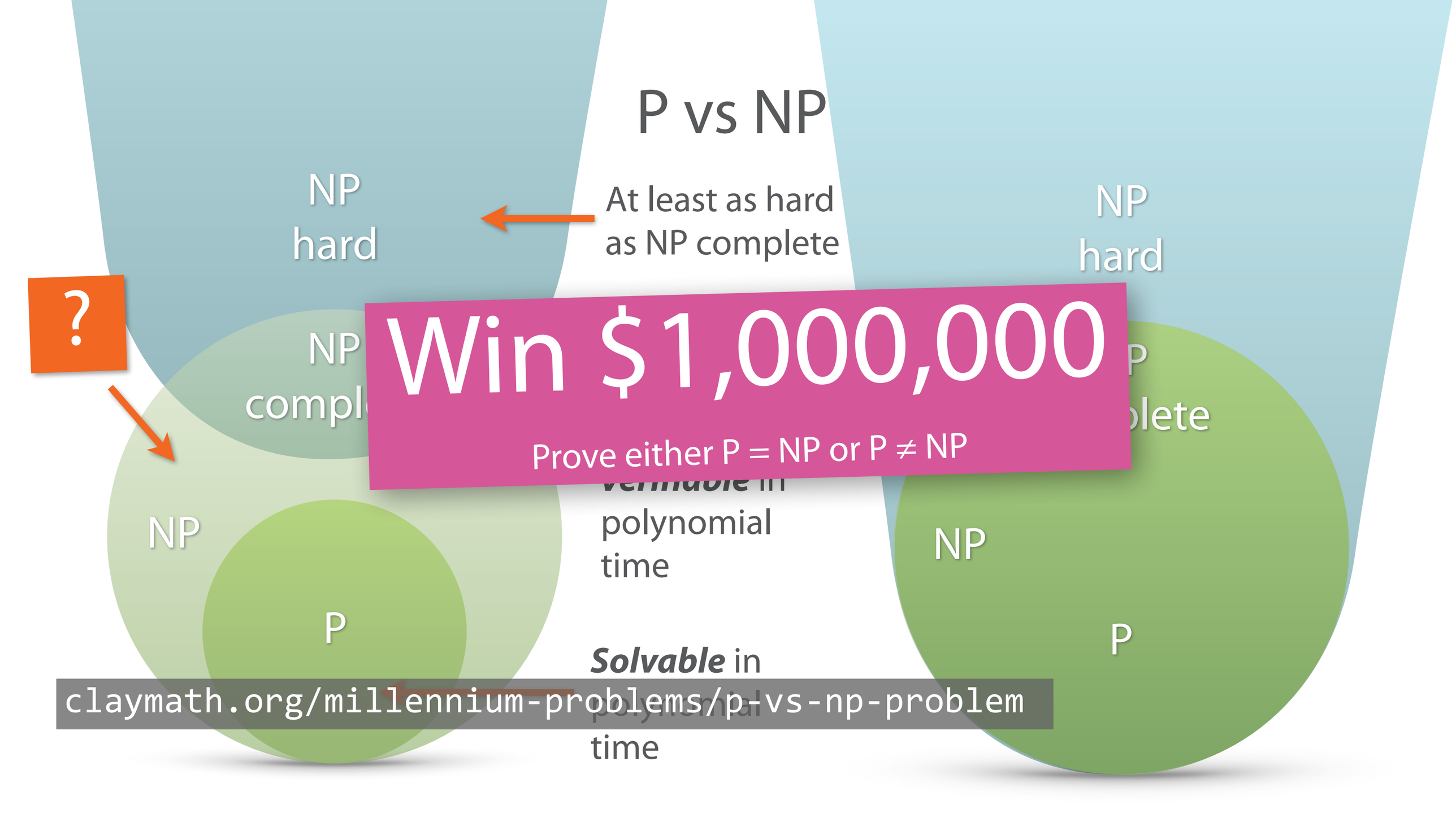

NP
hard
?
NP
complete
NP
P
At least as hard
as NP complete
The hardest in NP
Decision problems
verifiable in
polynomial
time
Solvable in
polynomial
time
NP
hard
NP
complete
NP
P

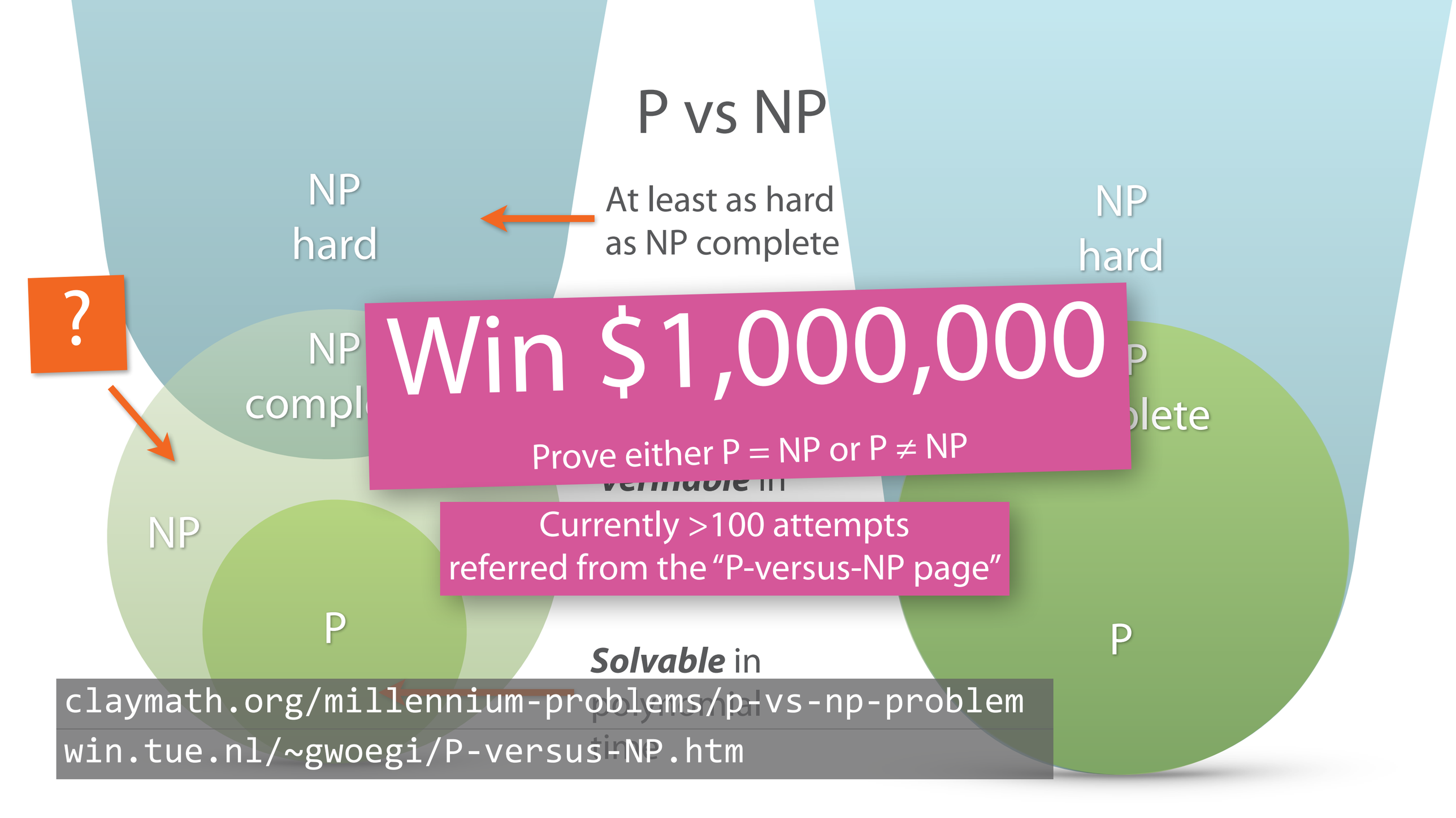
P vs NP


NP
hard
At least as hard
as NP complete
?
NP
complete
The hardest in NP
Decision problems
verifiable in
polynomial
time
NP
P
Solvable in
polynomial
time
NP
hard
NP
complete
NP
P

P vs NP


NP
hard
?
NP
complete
NP
P
At least as hard
as NP complete
The hardest in NP
Decision problems
verifiable in
polynomial
time
Solvable in
polynomial
time
NP
hard
NP
complete
NP
P


P vs NP
NP hard
At least as hard as NP complete
NP hard
?
Win $1,000,000
Prove either P = NP or P ≠ NP
NP
P
polynomial time
NP
P
claymath.org/millennium-problems/p-vs-np-problem


P vs NP
NP hard
At least as hard as NP complete
NP hard
?
Win $1,000,000
Prove either P = NP or P ≠ NP
Currently >100 attempts referred from the “P-versus-NP page”
NP
P
P
Solvable in
claymath.org/millennium-problems/p-vs-np-problem
win.tue.nl/~gwoegi/P-versus-NP.htm

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Hard Problems

Input size in #bits

P vs NP

Heuristics
and
approximation

Heuristics

Heuristics

Heuristics

Heuristics

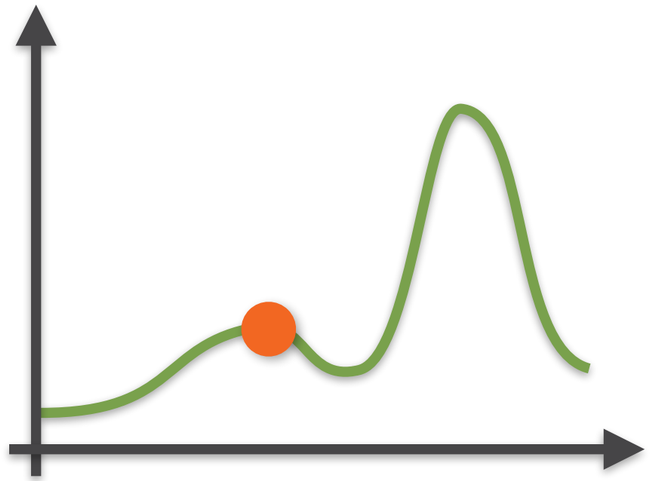
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?



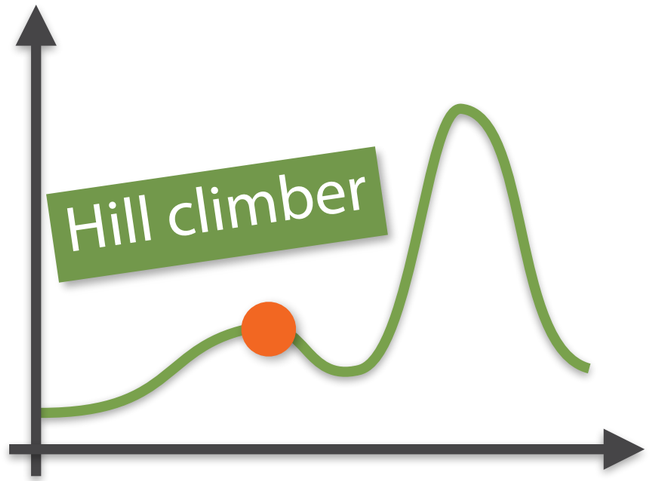
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?



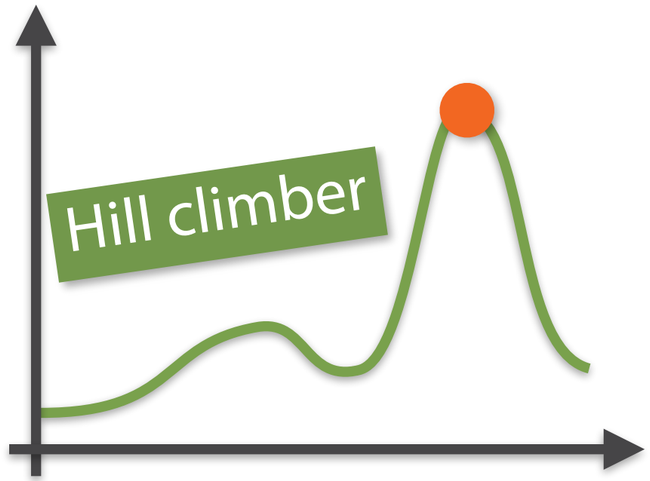
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?



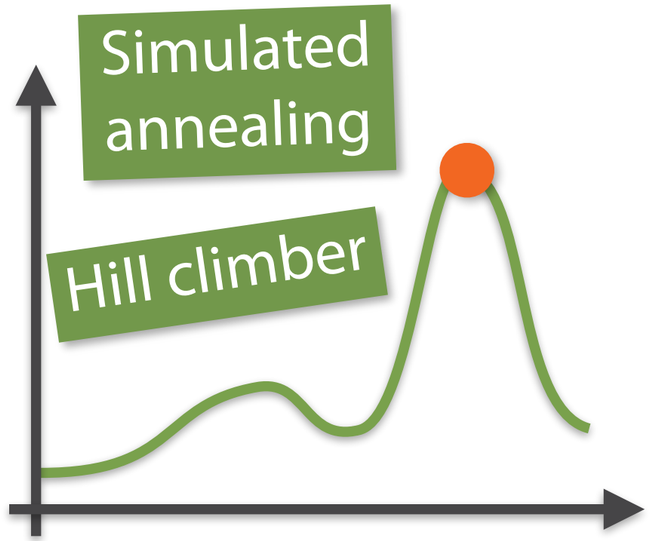
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?



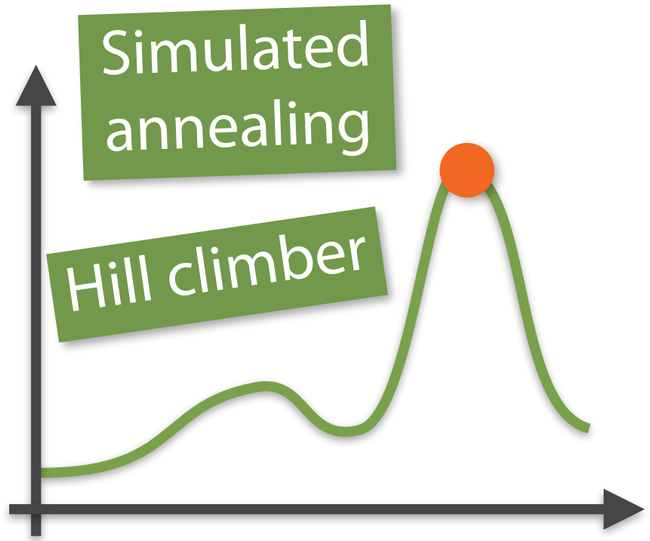
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?



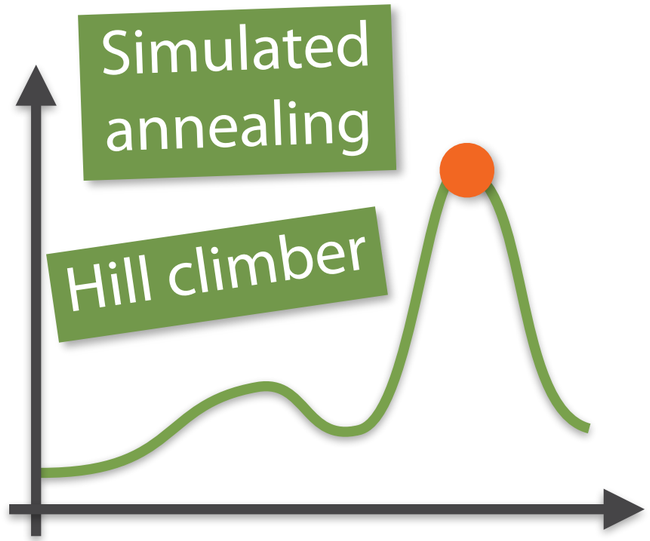
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?



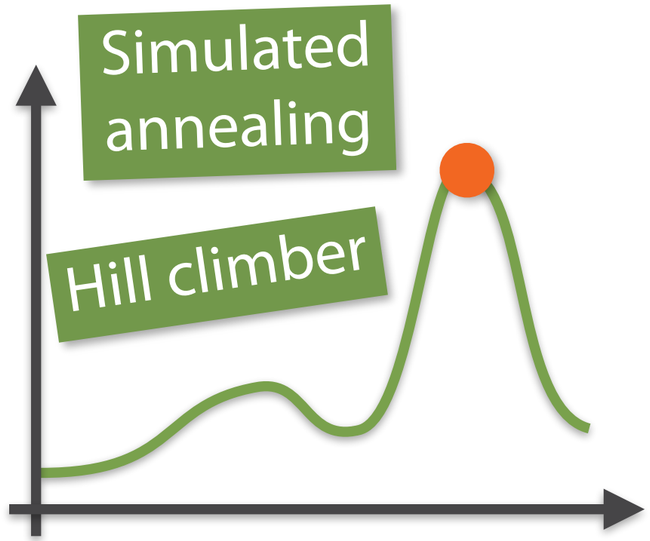
Heuristics

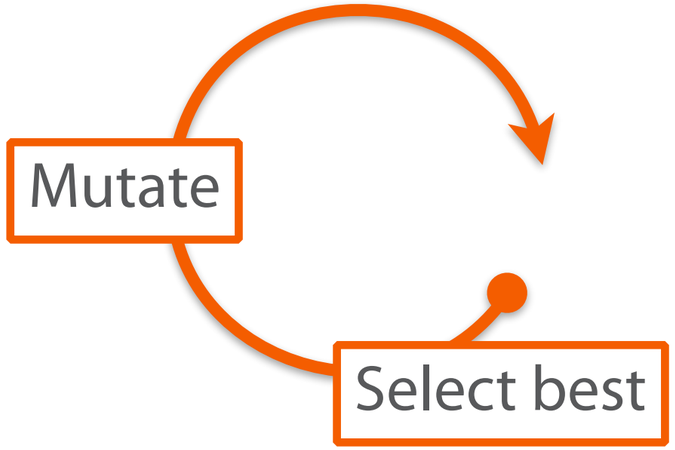
WANTED

Speed!

Trade with accuracy

Win simplicity?





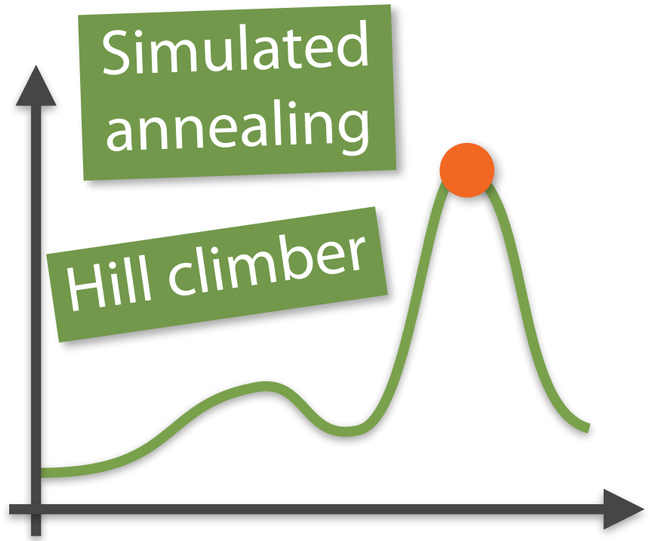
Heuristics

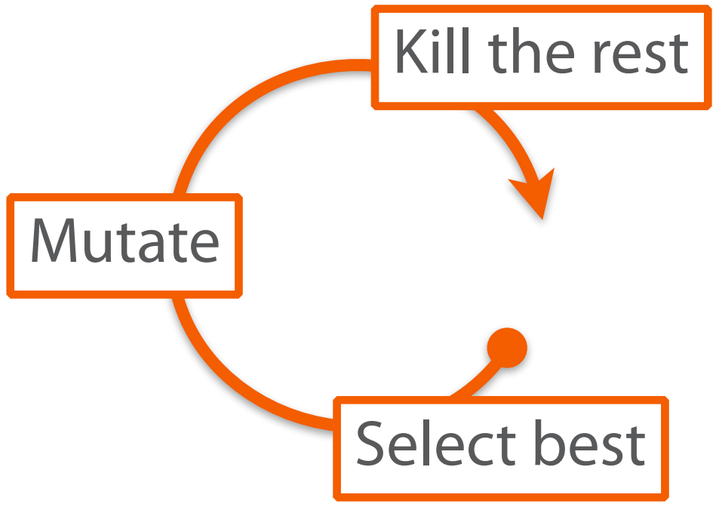
WANTED

Speed!

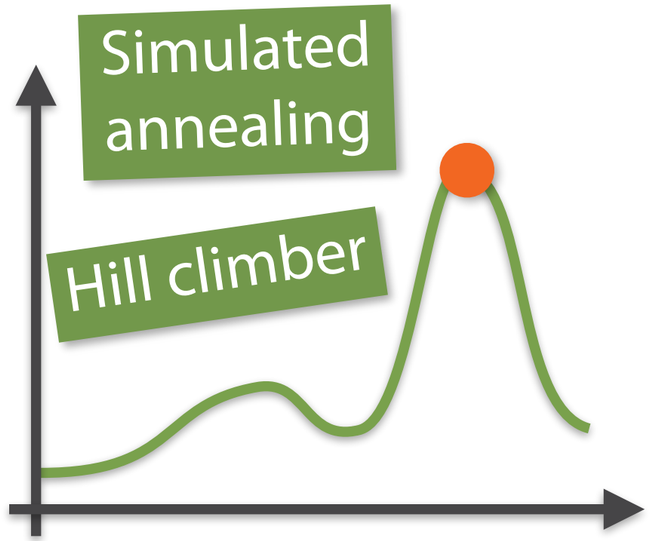
Trade with accuracy

Win simplicity?





Heuristics

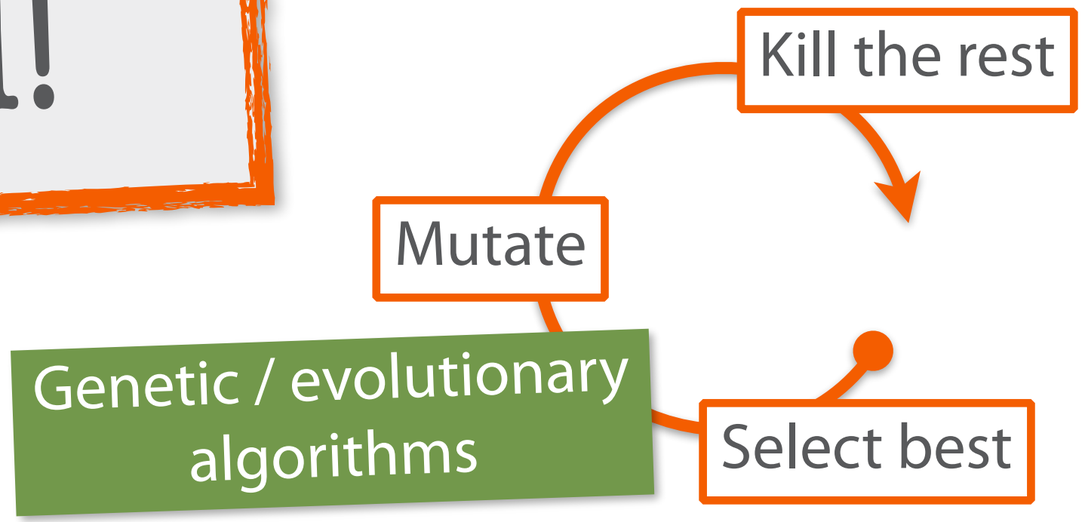


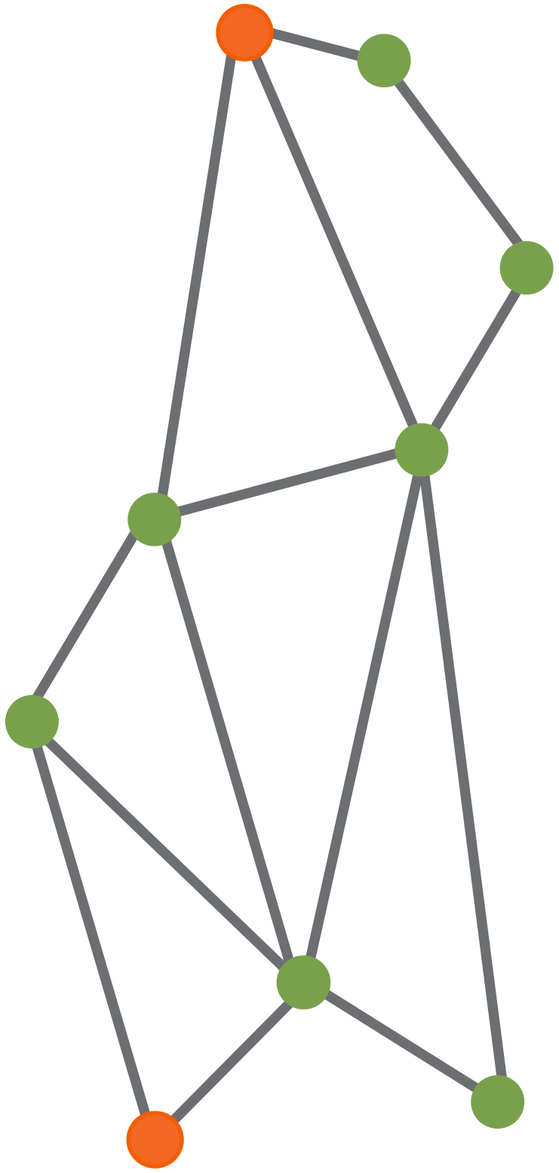
WANTED

Speed!

Trade with accuracy

Win simplicity?





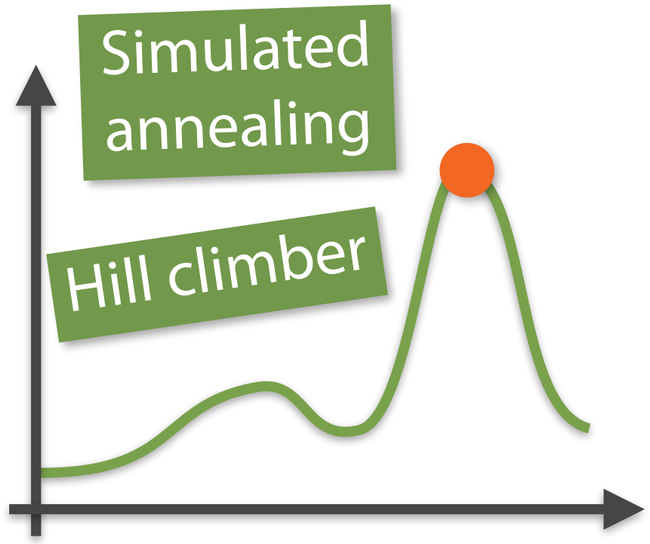
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?

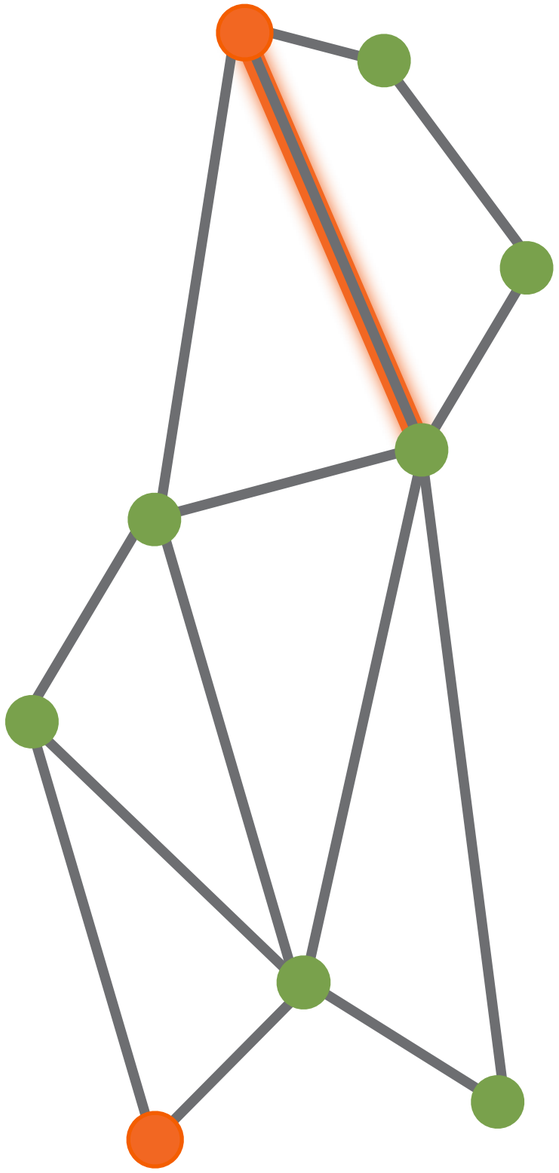


Kill the rest

Mutate

Genetic / evolutionary algorithms

Select best



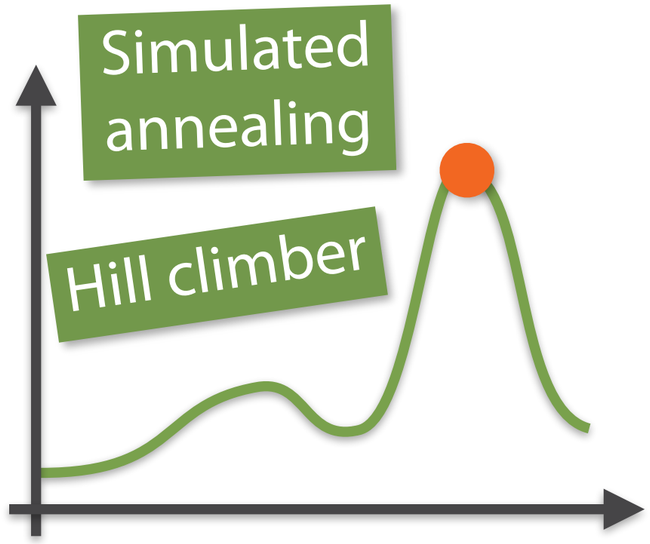
Heuristics

WANTED

Speed!

Trade with accuracy

Win simplicity?



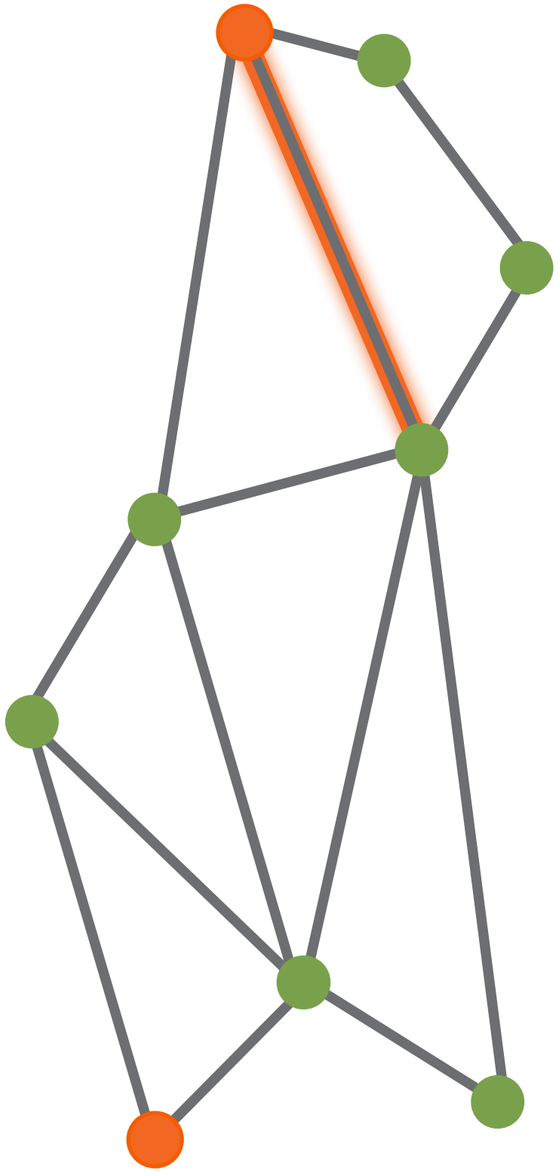
Kill the rest

Mutate

Genetic / evolutionary algorithms

Select best

Heuristics



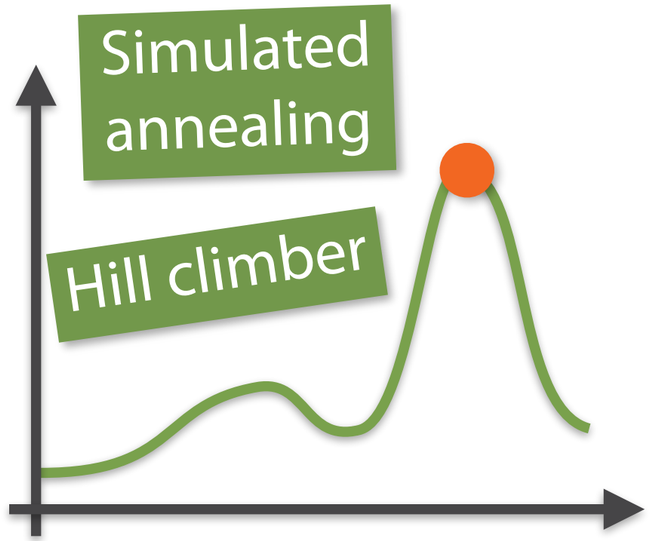
Distance ≥

WANTED

Speed!

Trade with accuracy

Win simplicity?



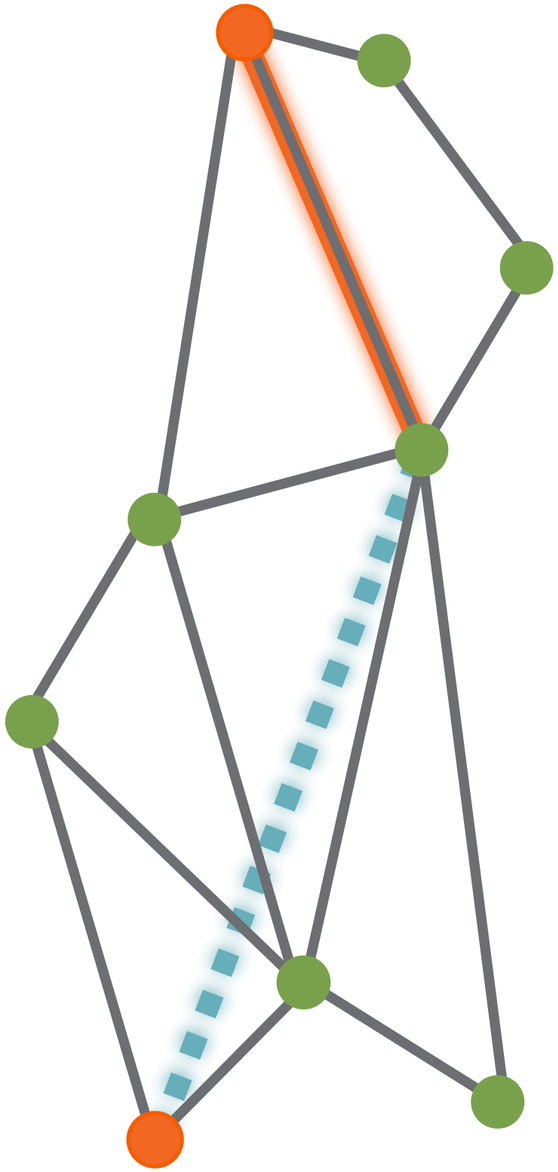
Genetic / evolutionary algorithms

Kill the rest

Mutate

Select best

Heuristics



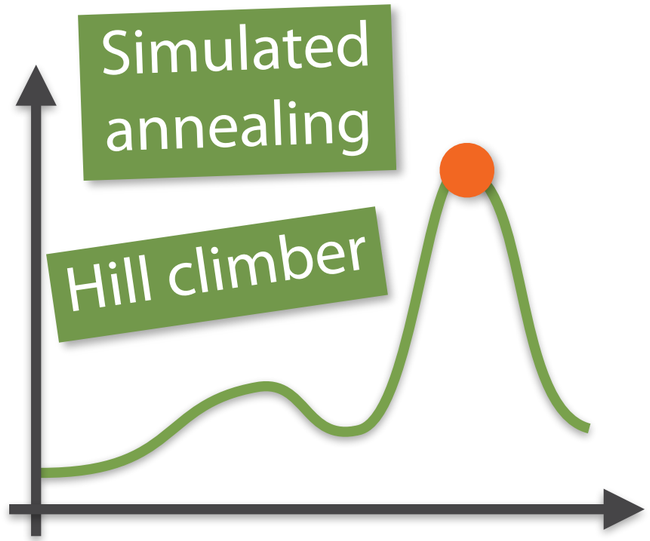
Distance ≥ +

WANTED

Speed!

Trade with accuracy

Win simplicity?



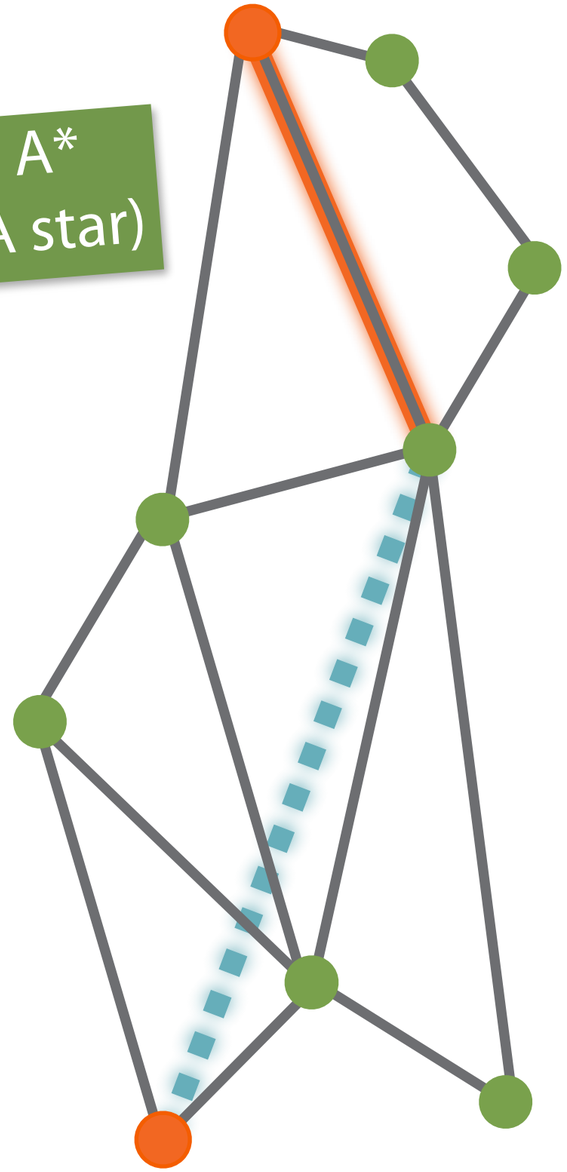
Genetic / evolutionary algorithms

Kill the rest

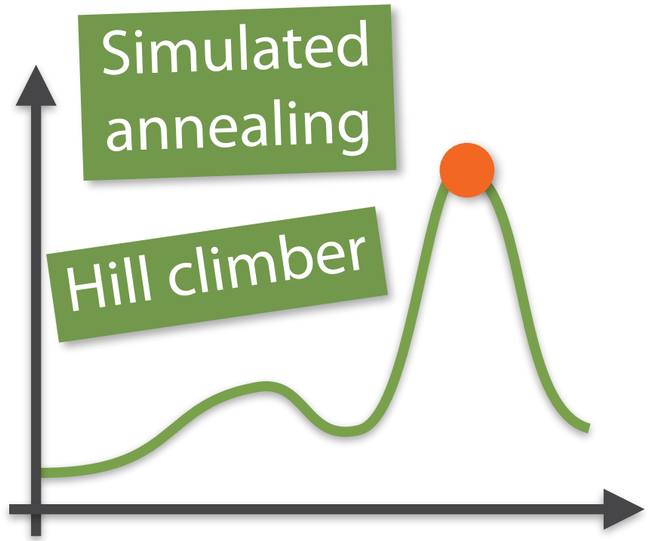
Mutate

Select best

Heuristics



Distance ≥ +

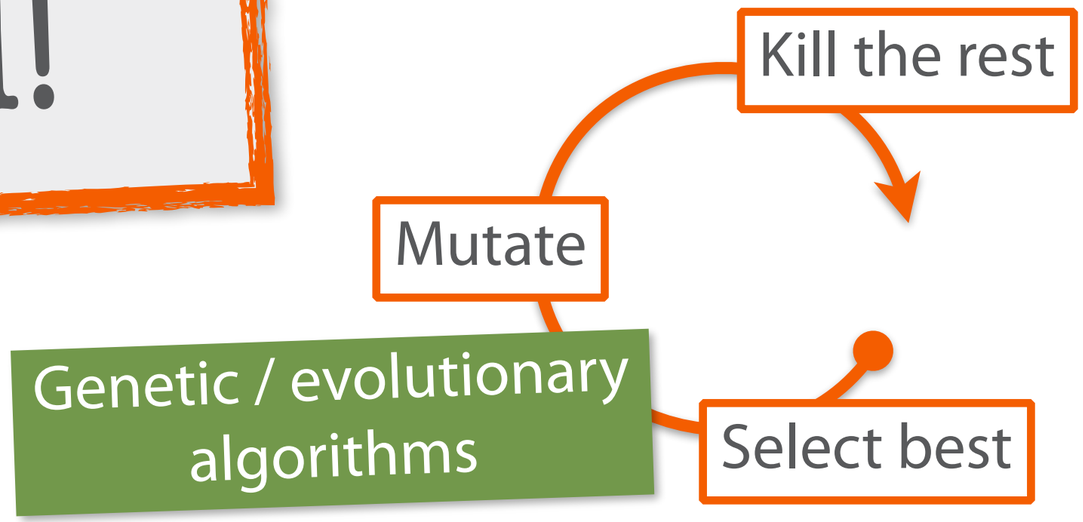


WANTED

Speed!

Trade with accuracy

Win simplicity?



Approximation Algorithms

Approximation Algorithms

- for optimization problems

Approximation Algorithms

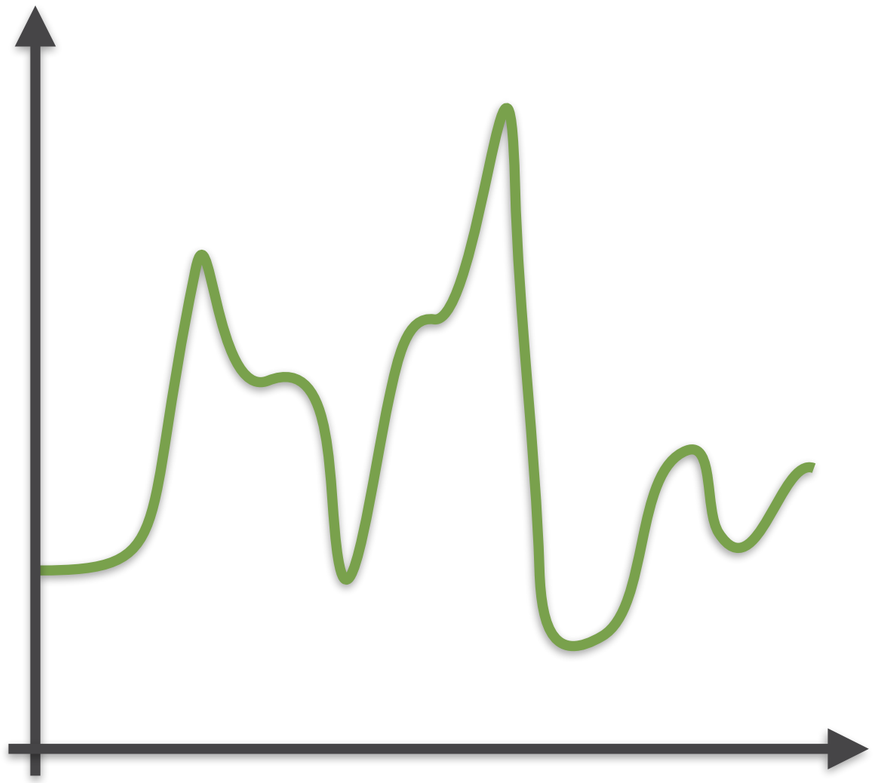
- for optimization problems

Runs in polynomial time

Approximation Algorithms

- for optimization problems

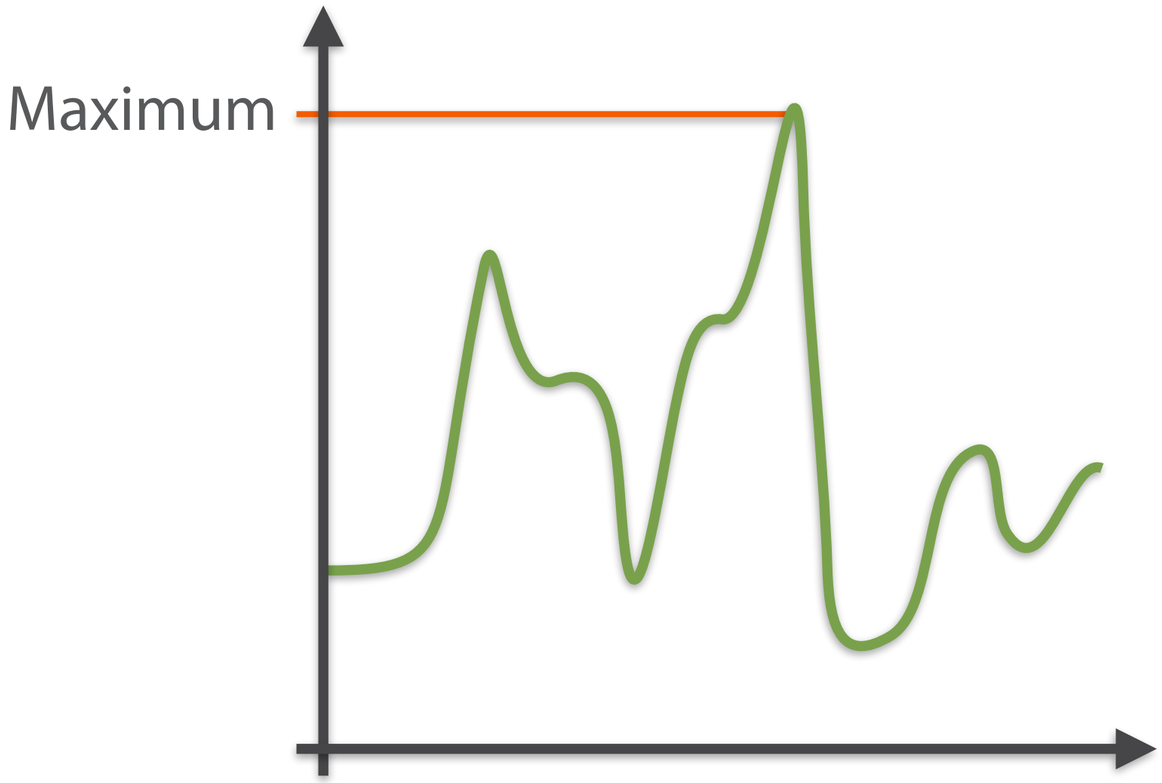
Runs in polynomial time



Approximation Algorithms

- for optimization problems

Runs in polynomial time

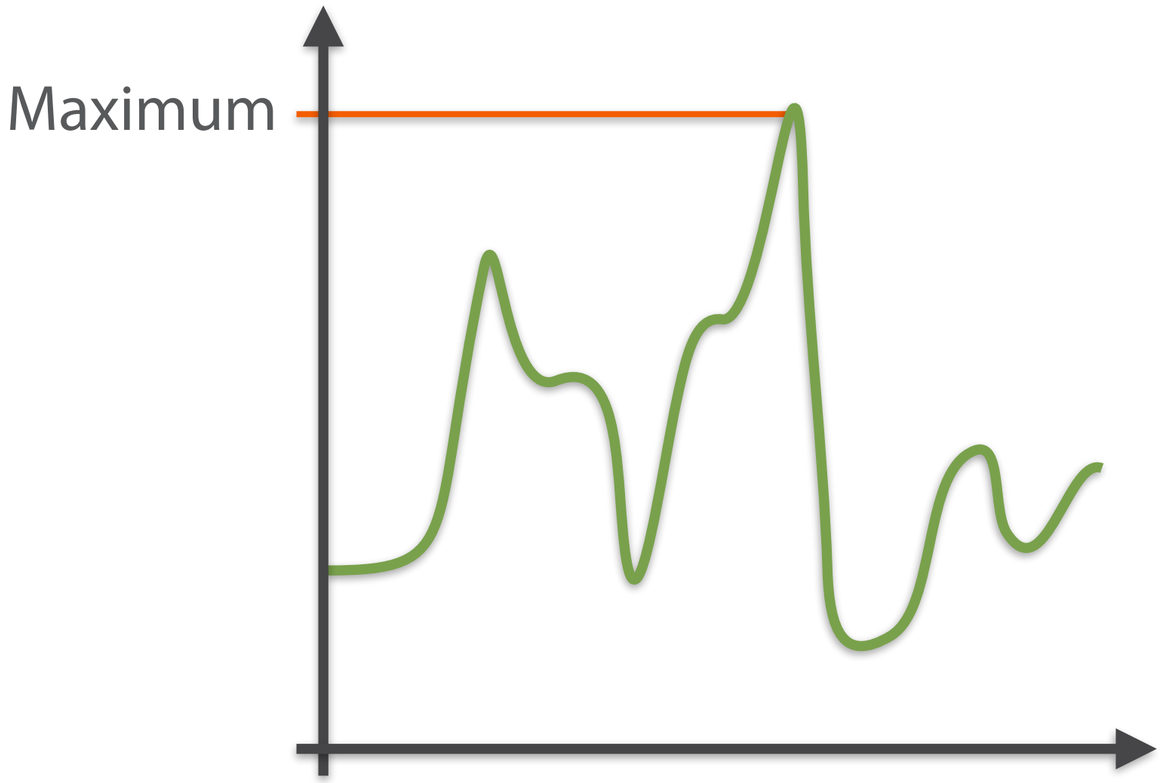


Approximation Algorithms

- for optimization problems

Runs in polynomial time

Allowed error: *p*

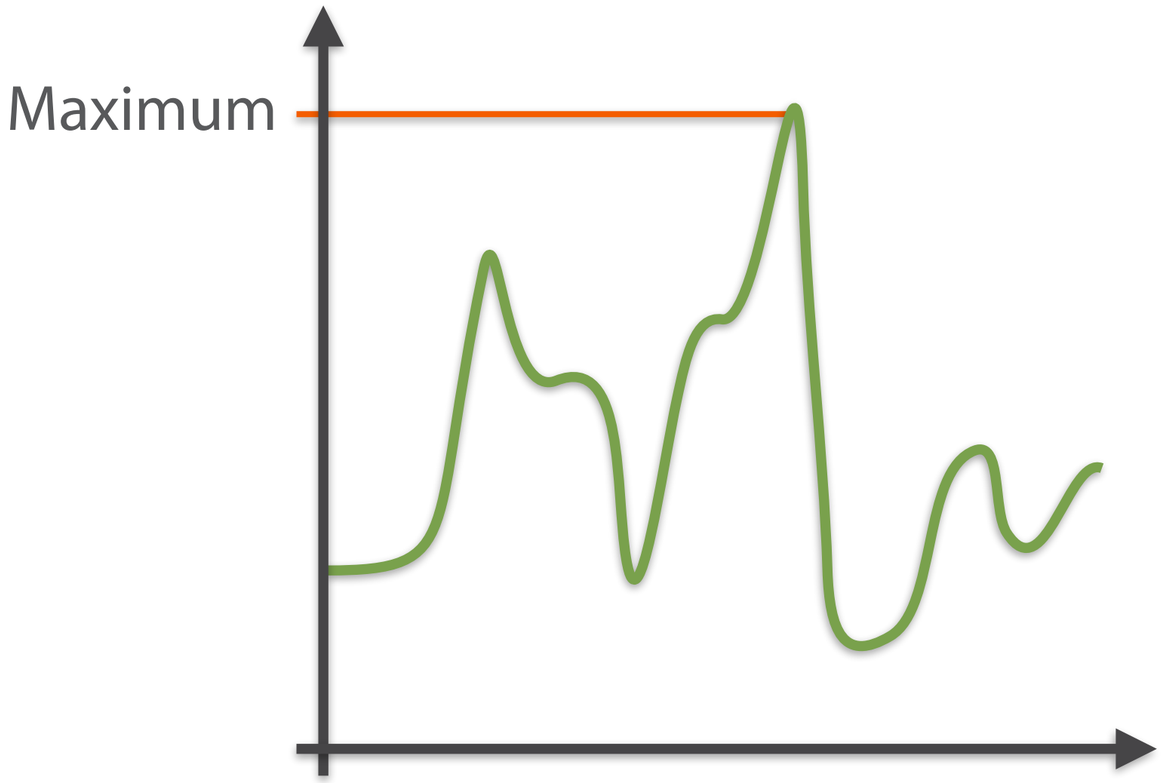


Approximation Algorithms

- for optimization problems

Runs in polynomial time

Allowed error: p $\quad 0 < p < 1$

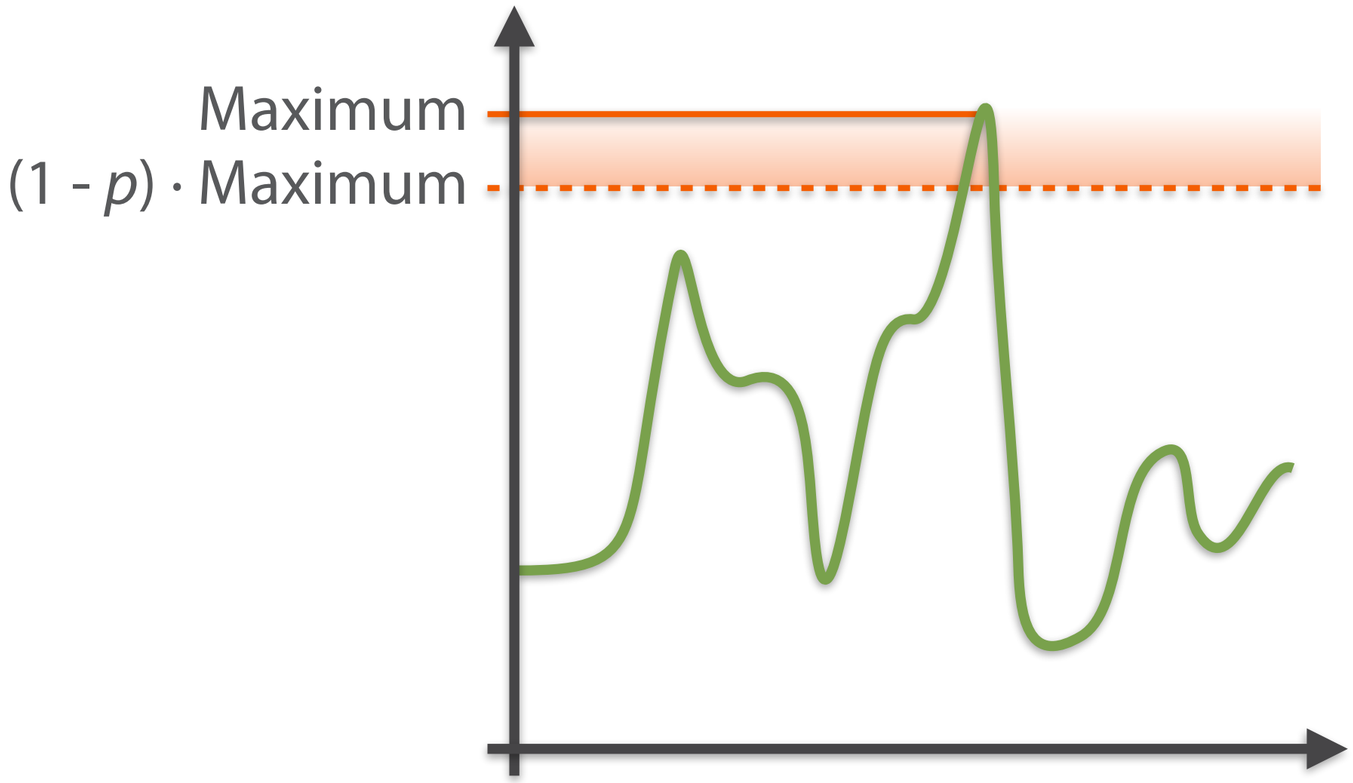


Approximation Algorithms

- for optimization problems

Runs in polynomial time

Allowed error: p $\quad 0 < p < 1$



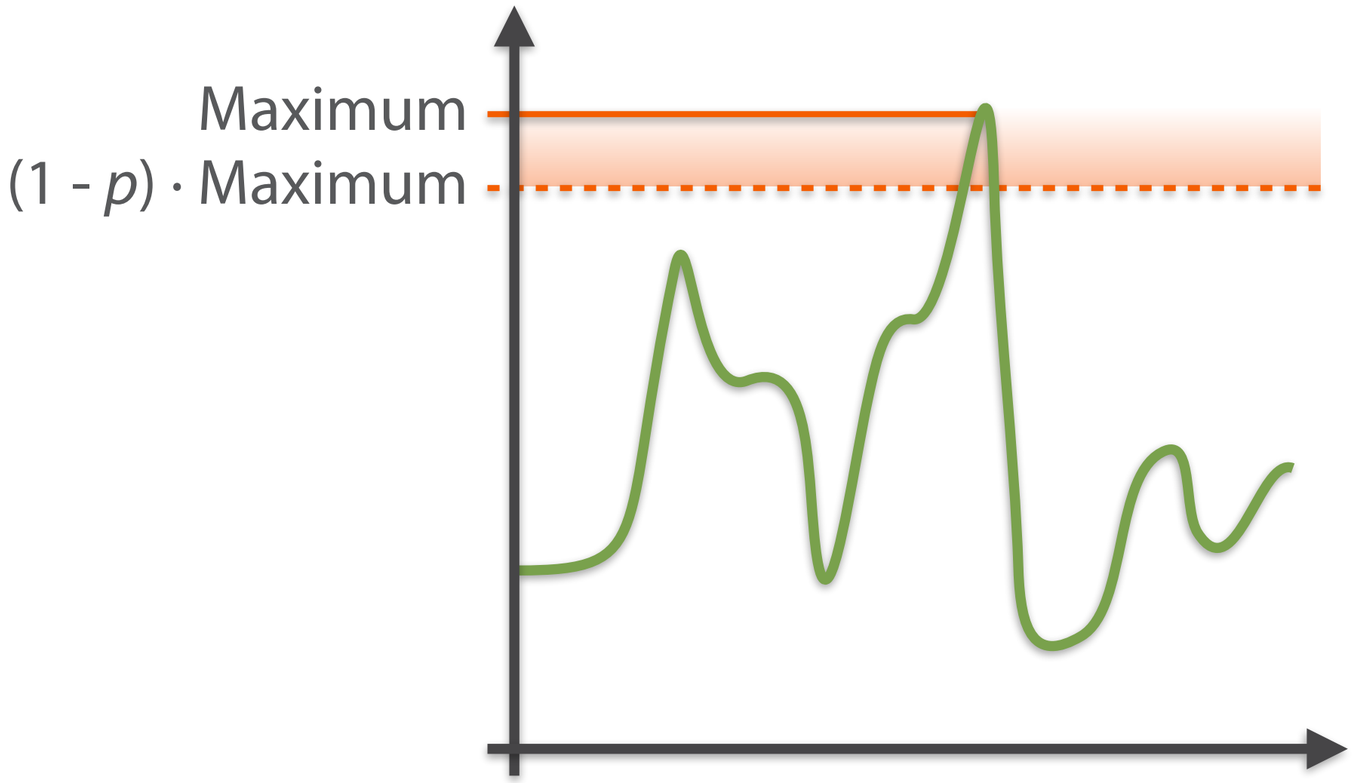
Approximation Algorithms

- for optimization problems

Runs in polynomial time

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



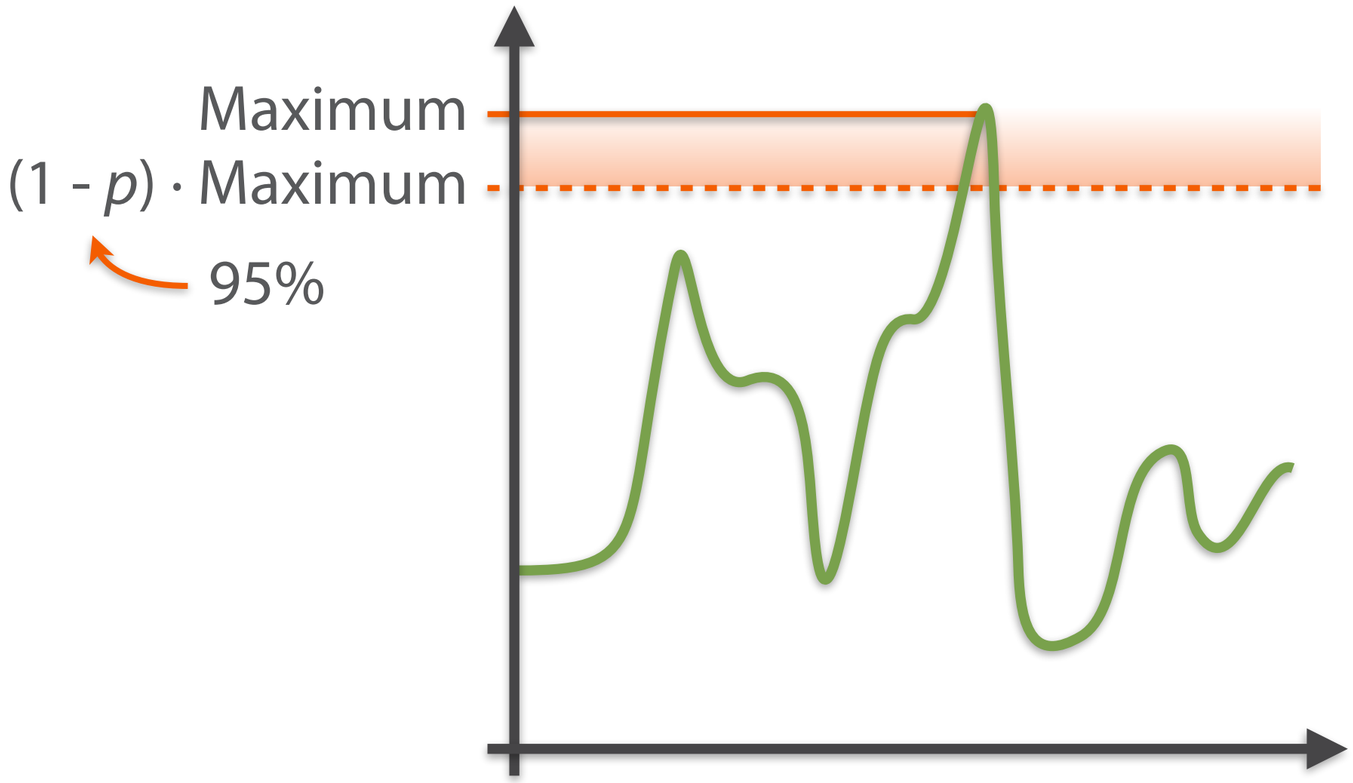
Approximation Algorithms

- for optimization problems

Runs in polynomial time

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



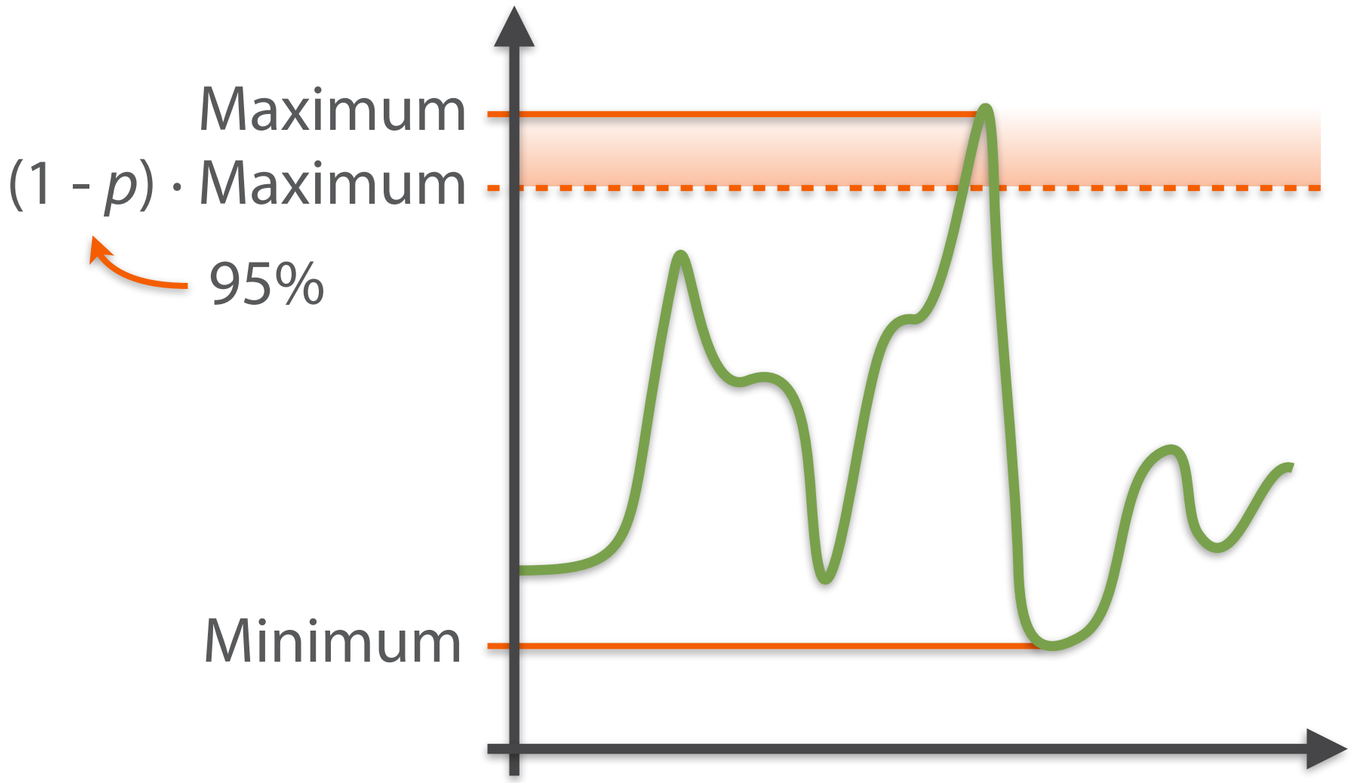
Approximation Algorithms

- for optimization problems

Runs in polynomial time

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



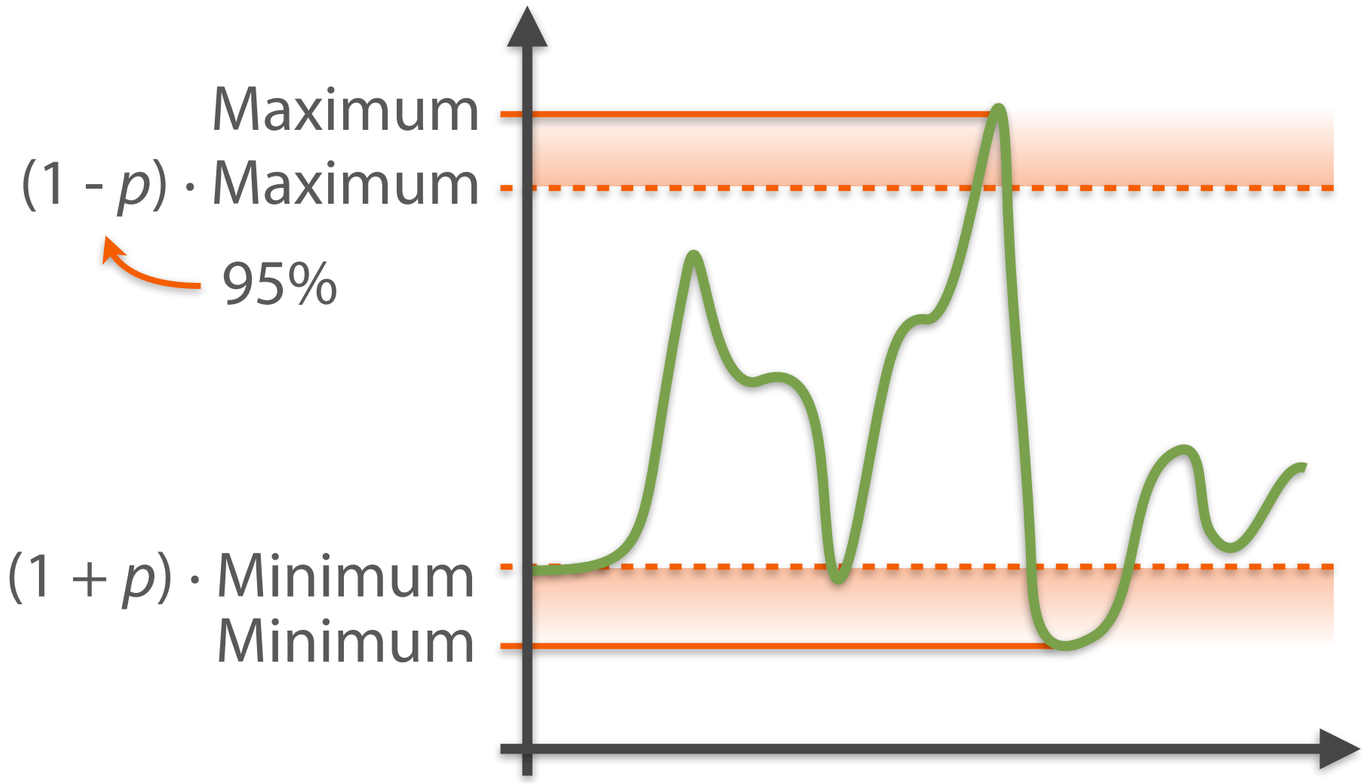
Approximation Algorithms

- for optimization problems

Runs in polynomial time

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



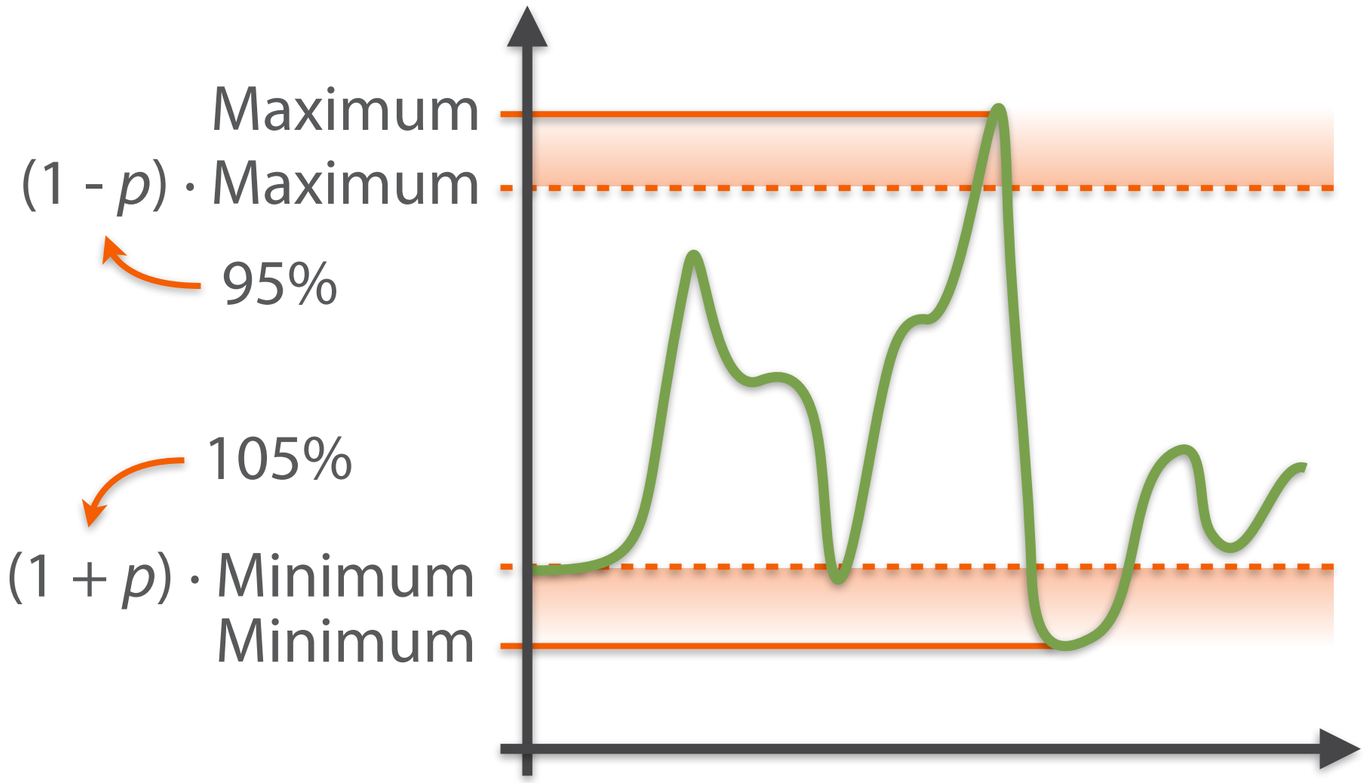
Approximation Algorithms

- for optimization problems

Runs in polynomial time

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



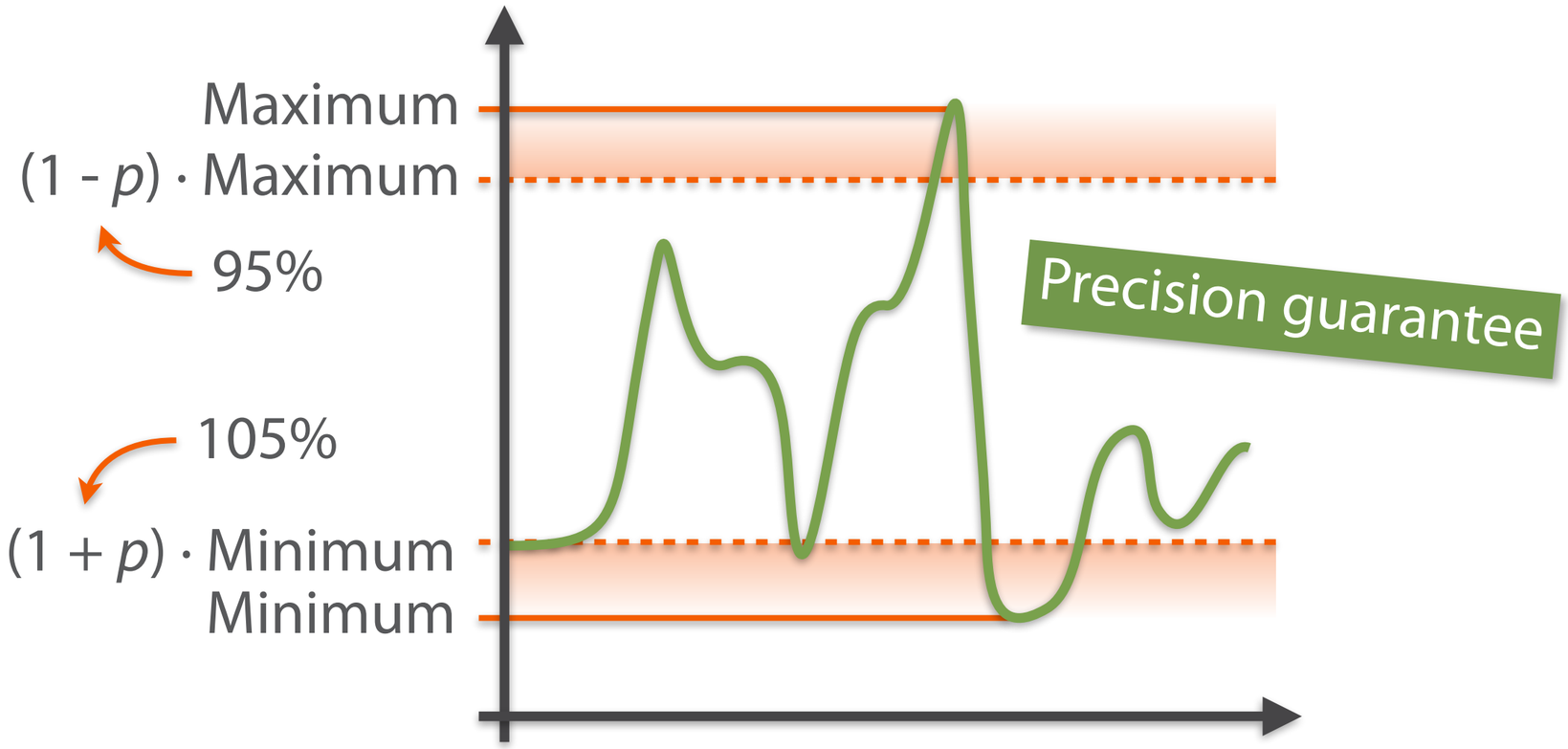
Approximation Algorithms

- for optimization problems

Runs in polynomial time

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



Approximation Algorithms

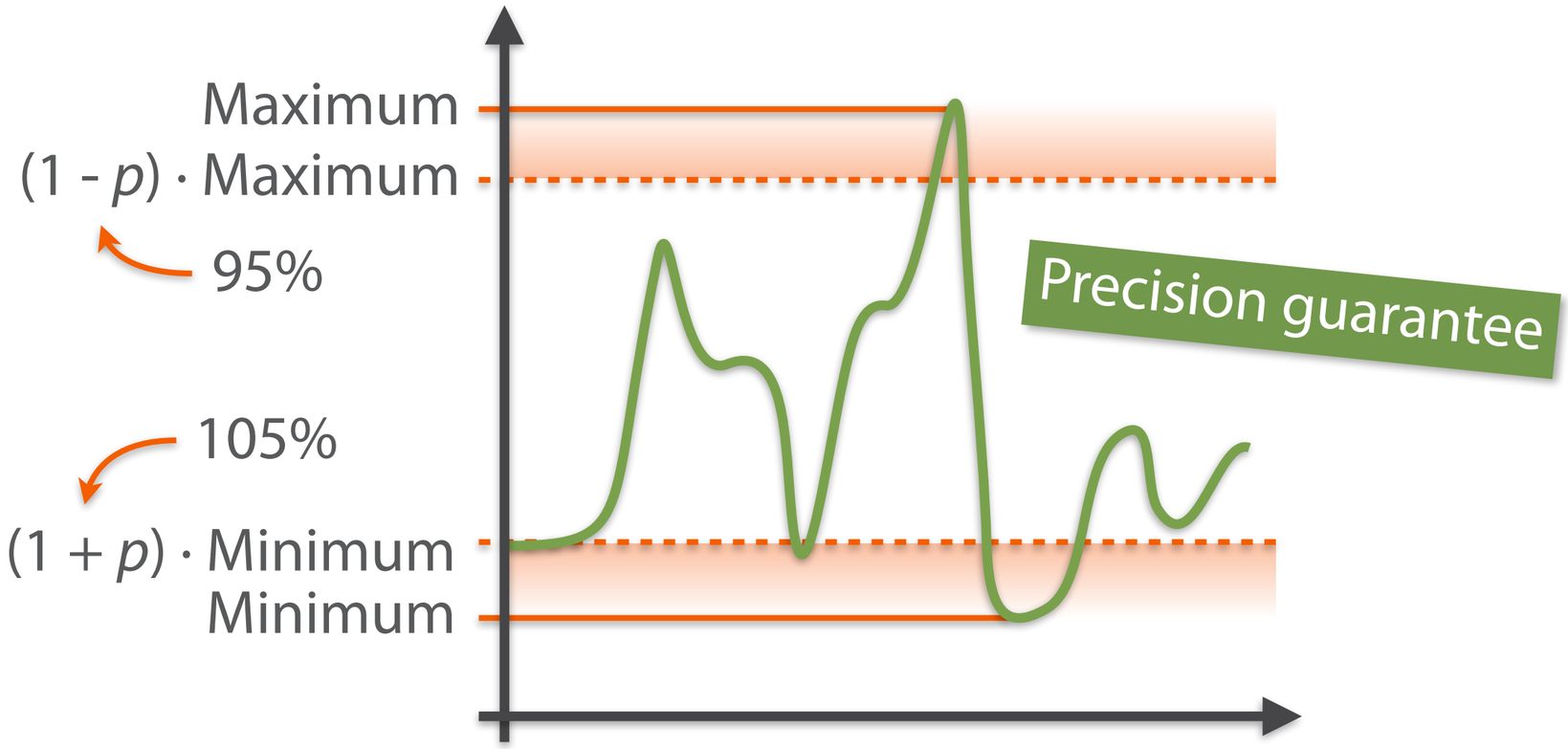
- for optimization problems

Runs in polynomial time

Polynomial-time approximation schemes (PTAS)

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



Approximation Algorithms

- for optimization problems

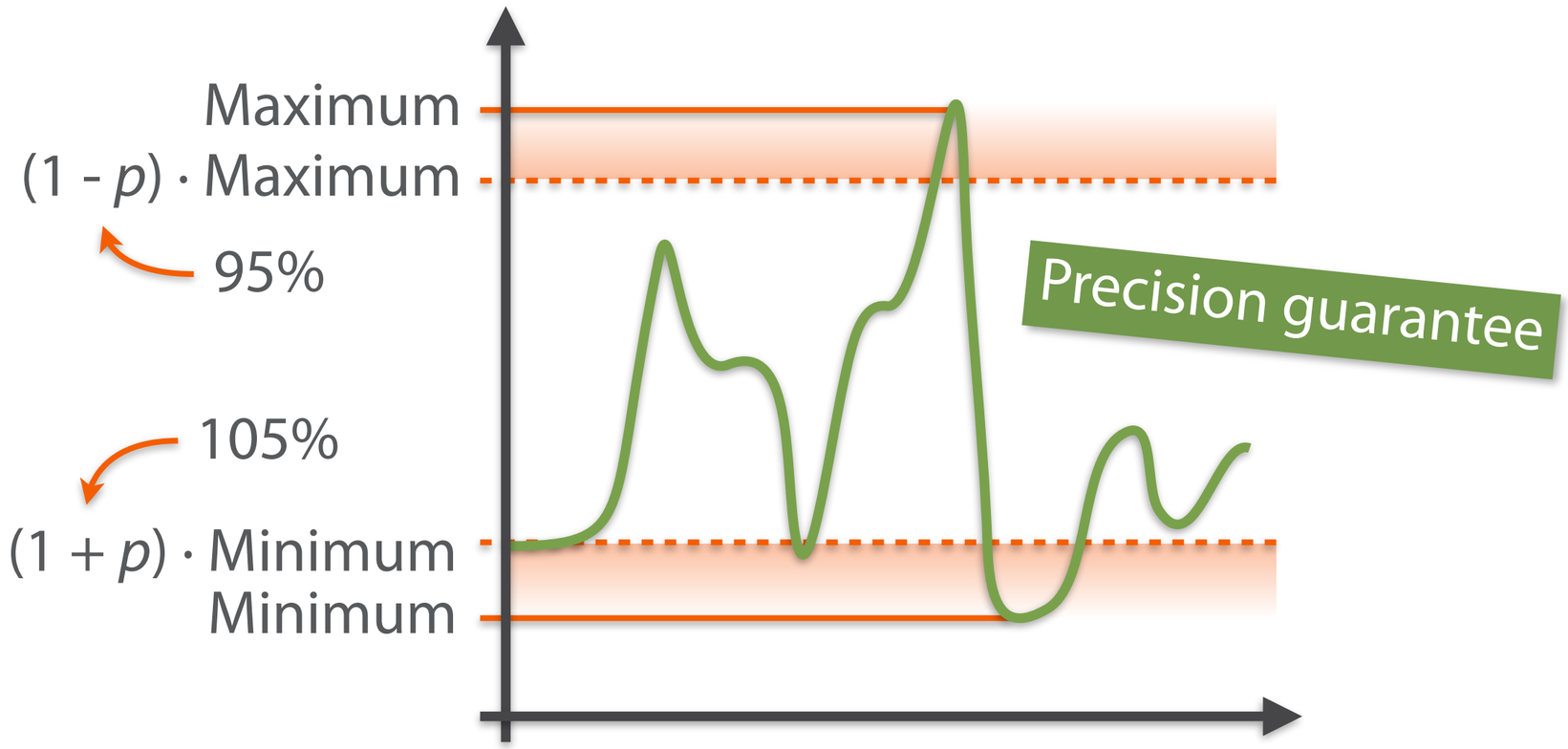
Runs in polynomial time

Polynomial-time approximation schemes (PTAS)

Complexity depends on error

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



Approximation Algorithms

- for optimization problems

Runs in polynomial time

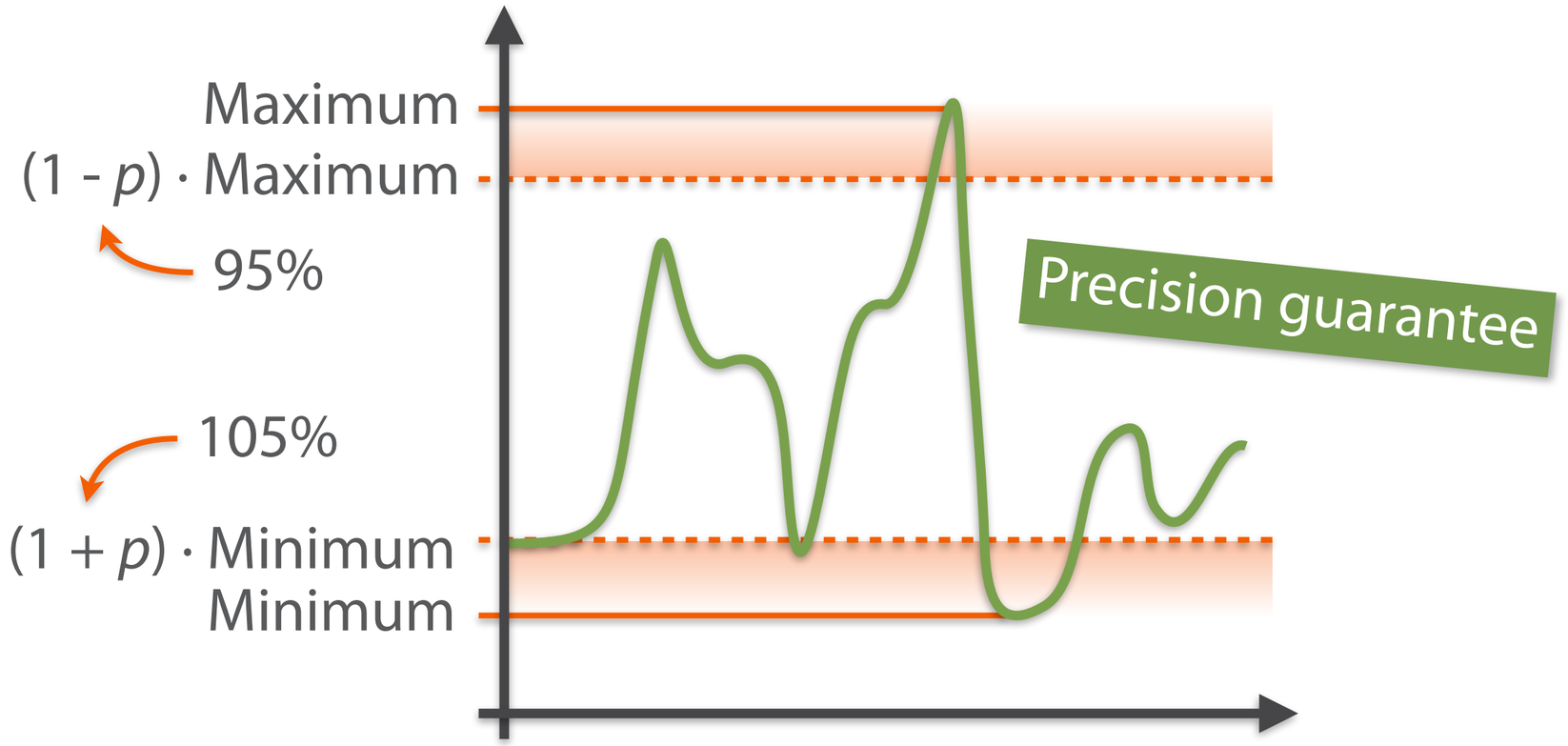
Polynomial-time approximation schemes (PTAS)

Complexity depends on error

$O(N^{1/p})$

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



Approximation Algorithms

- for optimization problems

Runs in polynomial time

Polynomial-time approximation schemes (PTAS)

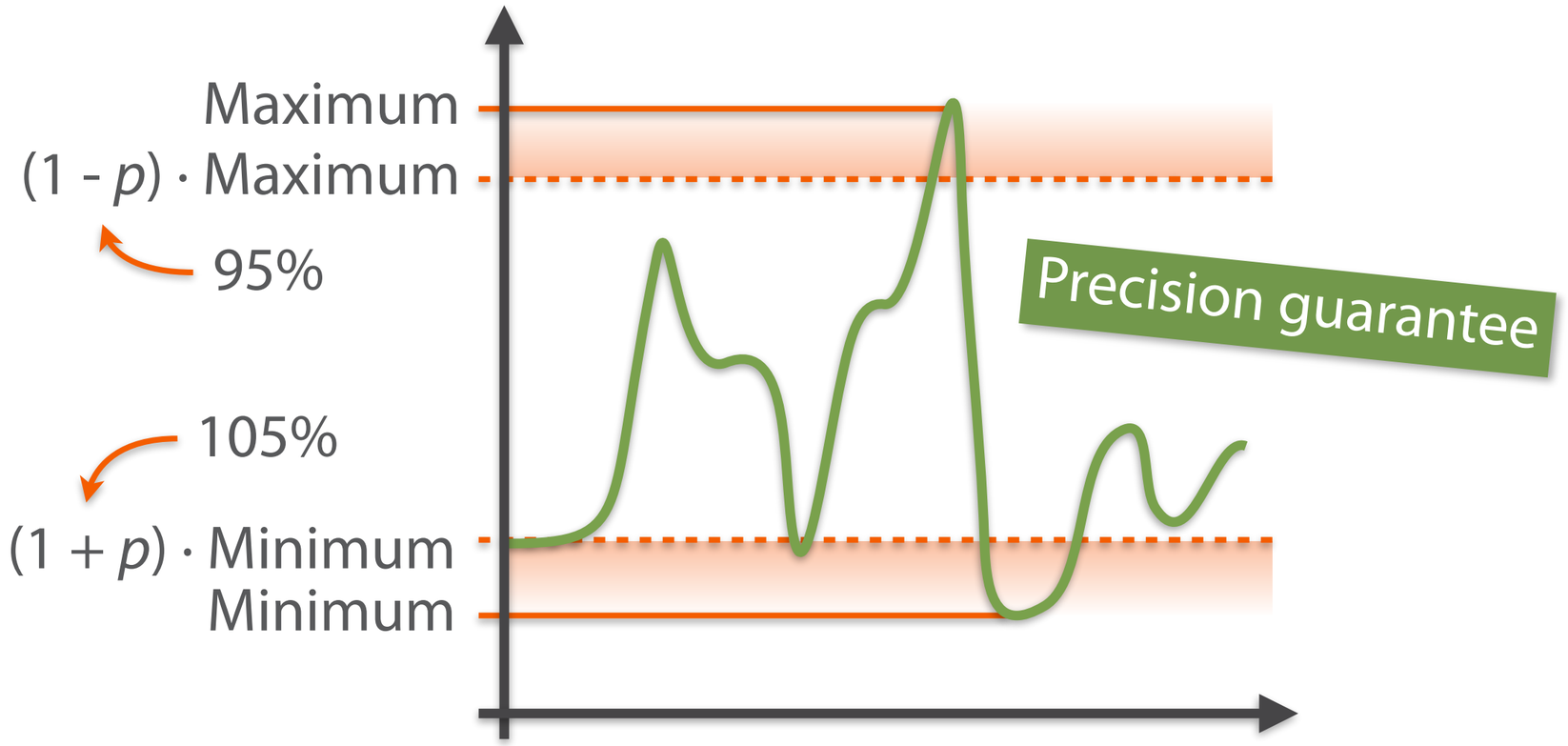
Complexity depends on error

$O(N^{1/p})$

$p = 0.1$: $O(N^{10})$

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



Approximation Algorithms

- for optimization problems

Runs in polynomial time

Polynomial-time approximation schemes (PTAS)

Complexity depends on error

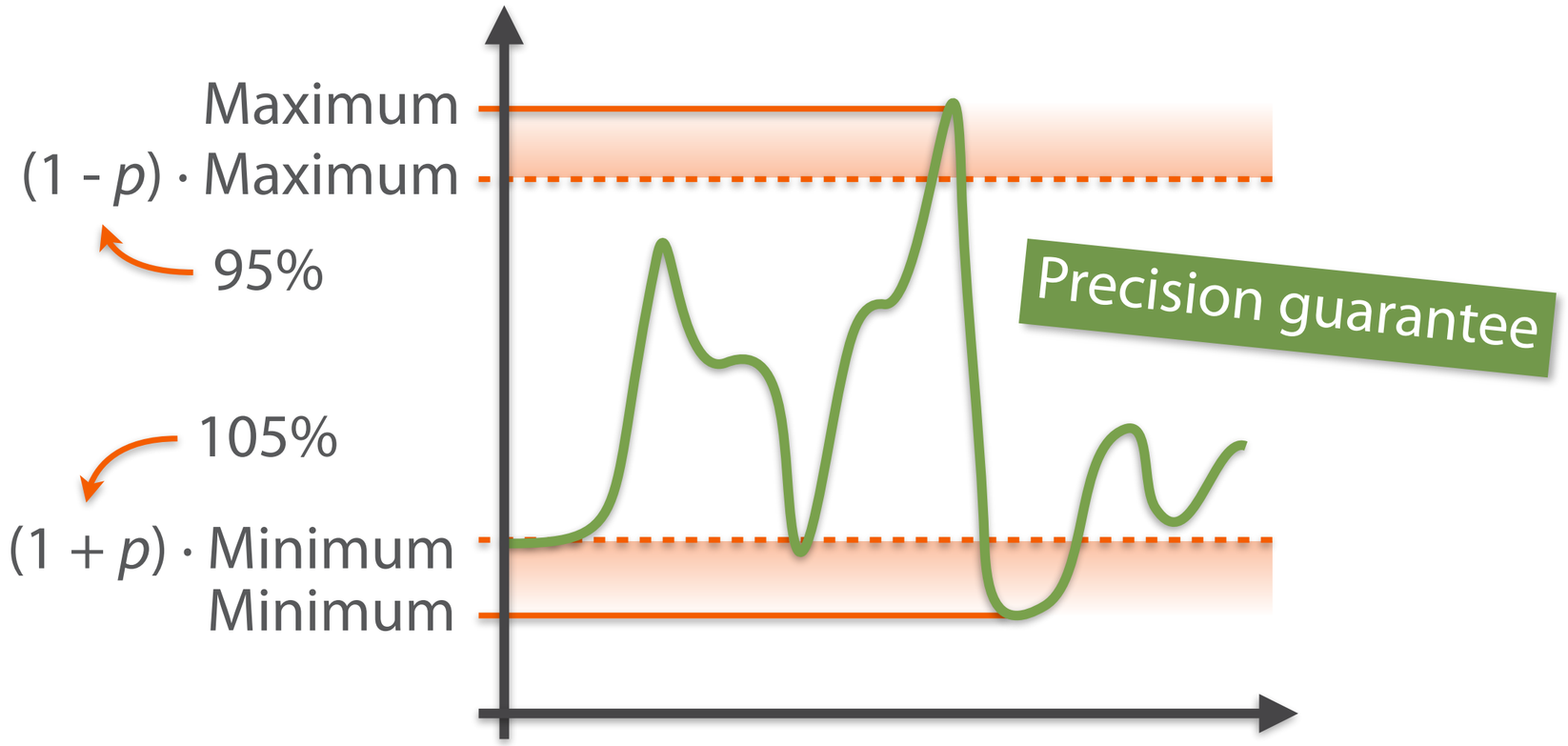
$O(N^{1/p})$

$p = 0.1$: $O(N^{10})$

$p = 0.05$: $O(N^{20})$

if $p = 0.05$

Allowed error: p $\quad 0 < p < 1$



Lessons Learned

Input size in #bits

P vs NP

Heuristics
and
approximation

Lessons Learned

Input size in #bits

P vs NP

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Input size in #bits

P vs NP

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P vs NP

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P vs NP

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

P vs NP

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

P = NP unsolved!

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

P = NP unsolved!

Heuristics
and
approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

P = NP unsolved!

Exchange precision for speed

Heuristics and approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

P = NP unsolved!

Exchange precision for speed

Heuristics: Generally no worst-case

Heuristics and approximation

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

P = NP unsolved!

Exchange precision for speed

Heuristics: Generally no worst-case

Heuristics and approximation

Approximation: Guaranteed quality

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

P = NP unsolved!

Heuristics and approximation

Exchange precision for speed

Heuristics: Generally no worst-case

Approximation: Guaranteed quality

Speed may depend on error

Lessons Learned

Complexity may cheat

Don’t overthink it

Input size in #bits

P: Solvable in polynomial time

NP: Verifiable in polynomial time

P vs NP

NP-hard: Generally awfully hard

P = NP unsolved!

Exchange precision for speed

Heuristics: Generally no worst-case

Heuristics and approximation

Approximation: Guaranteed quality

Speed may depend on error