Reduction of NP Problems
&
Property-Based Testing

Chenhao Zhang

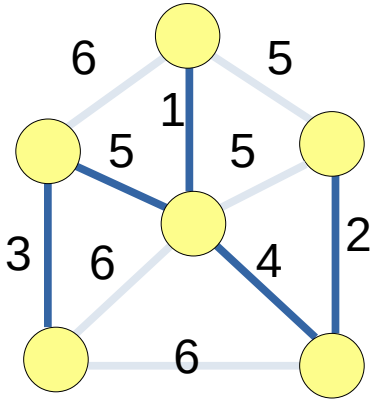
CS396 Fall 2023
Northwestern

Plan of the week

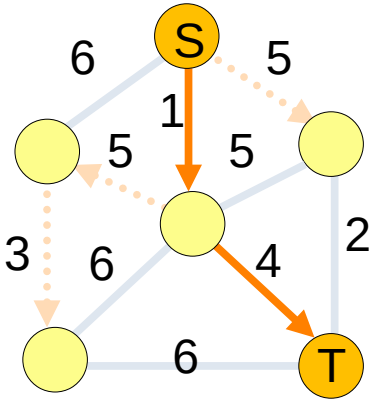
- **NP Problem & Reduction (Today)**
- Examples, Reduction in Karp -- Wednesday
- Lab, Assignment 5 -- Friday

Many problems have efficient algorithms

Minimum Spanning Tree



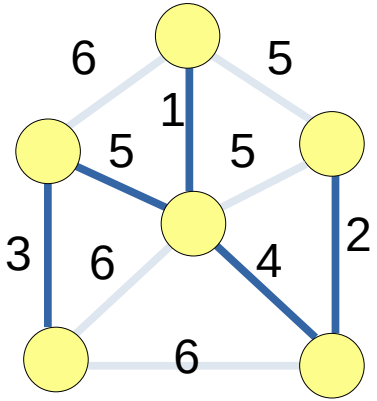
Shortest path



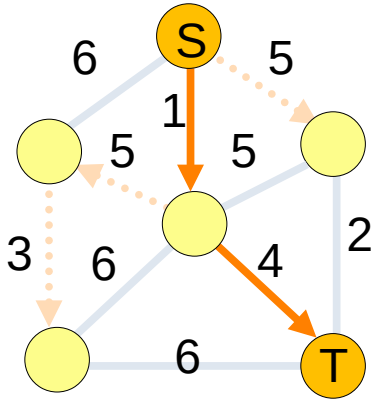
.....

Many problems have efficient algorithms

Minimum Spanning Tree



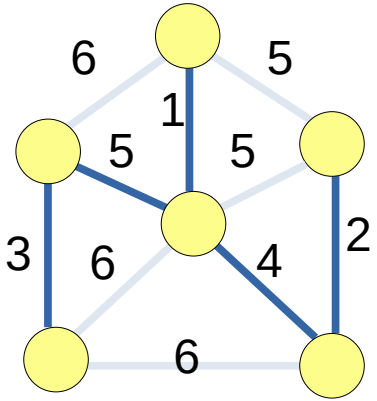
Shortest path



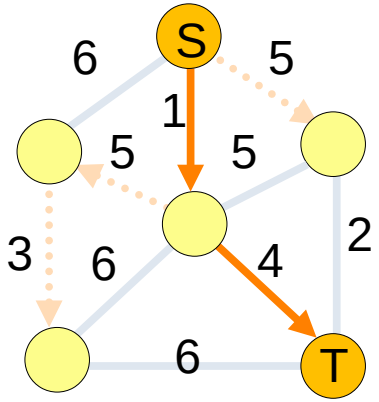
.....

version with Yes/No answer

Has Spanning Tree w/ Cost <=15 ?



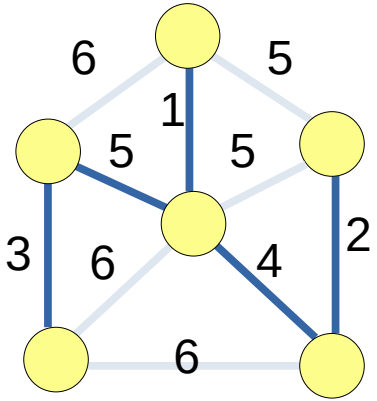
Has S-T path w/ Cost <=5 ?



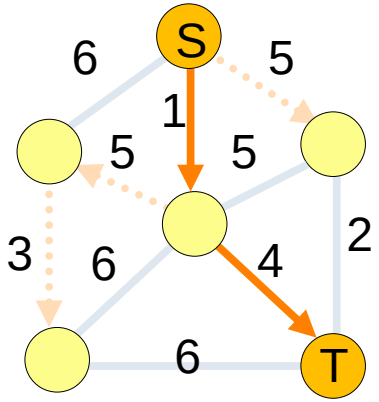
.....

version with Yes/No answer – *decision problem*

Has Spanning Tree w/ Cost <=15 ?



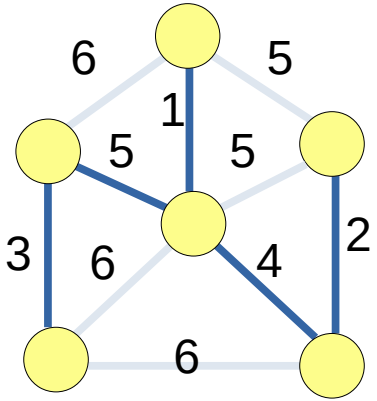
Has S-T path w/ Cost <=5 ?



.....

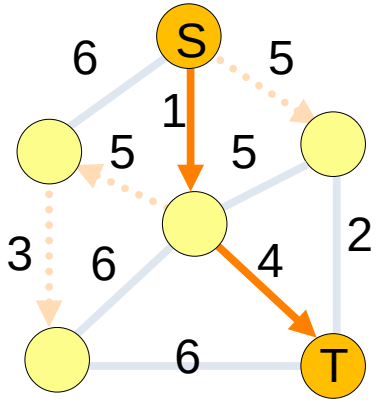
version with Yes/No answer – *decision problem*

Has Spanning Tree w/ Cost <=15 ?



1+5+3+4+2=15

Has S-T path w/ Cost <=5 ?

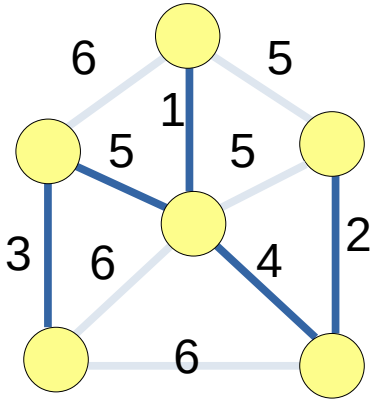


1+4=5

. . . . . .

version with Yes/No answer – *decision problem*

Has Spanning Tree w/ Cost <=15 ?

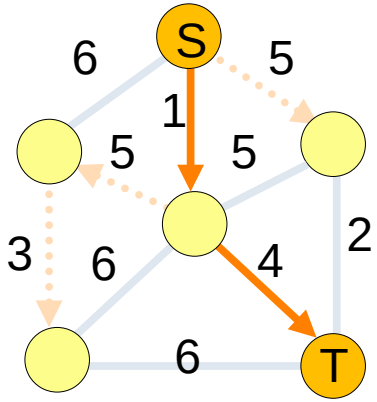


Yes

1+5+3+4+2=15

.....

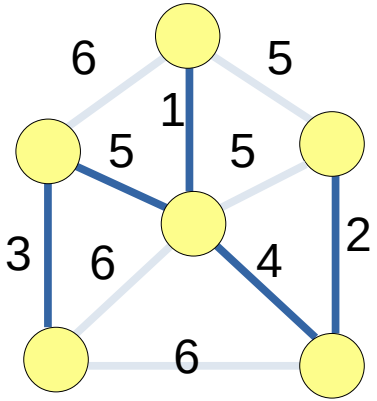
Has S-T path w/ Cost <=5 ?



1+4=5

version with Yes/No answer – *decision problem*

Has Spanning Tree w/ Cost <=15 ?

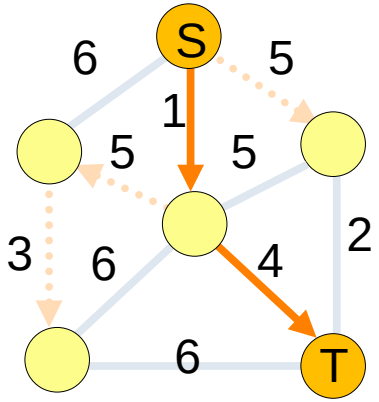


Yes

1+5+3+4+2=15

.....

Has S-T path w/ Cost <=5 ?

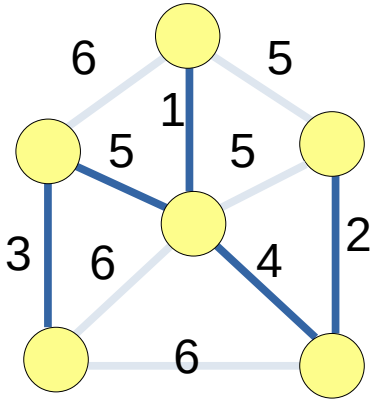


Yes

1+4=5

Yes-Instance has a *certificate*, i.e., proof of yes

Has Spanning Tree w/ Cost <=15 ?

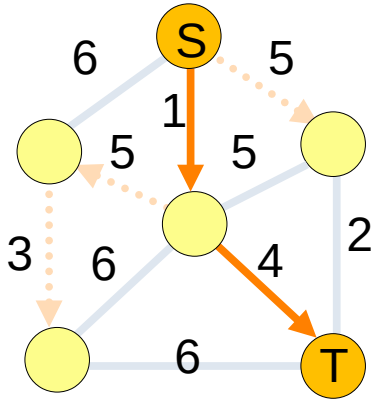


Yes

1+5+3+4+2=15

.....

Has S-T path w/ Cost <=5 ?

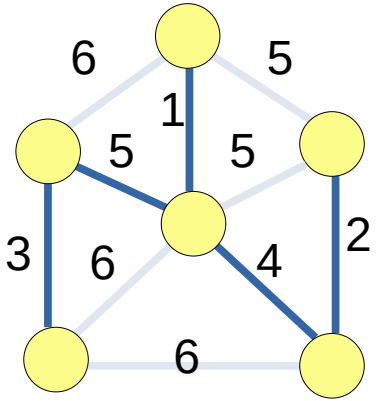


Yes

1+4=5

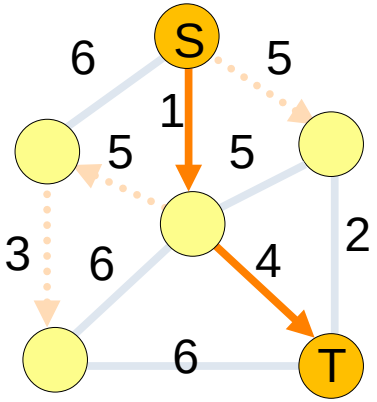
No-Instance has no *certificate*, ~~proof of yes~~

Has Spanning Tree w/ Cost <=**14** ?



1+5+3+4+2=15 > 14

Has S-T path w/ Cost <=**4** ?

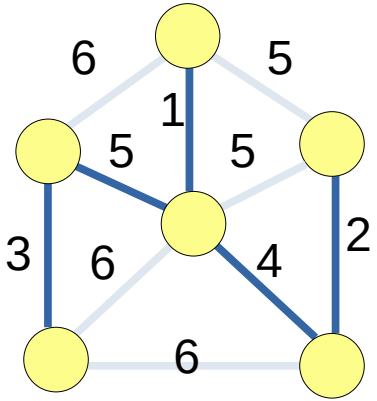


1+4=5 > 4

. . . . . .

No-Instance has no *certificate*, ~~proof of yes~~

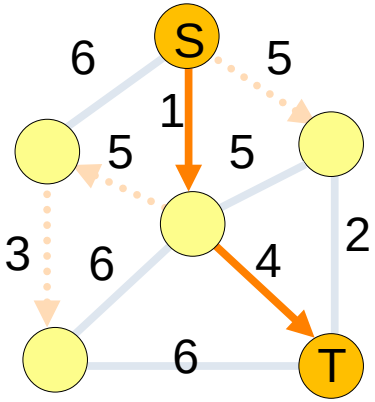
Has Spanning Tree w/ Cost <=**14** ?



✘

1+5+3+4+2=15 > 14

Has S-T path w/ Cost <=**4** ?

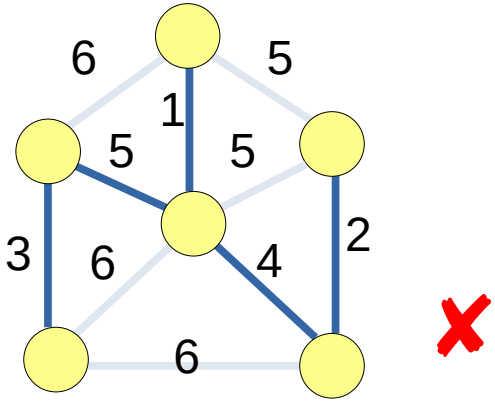


1+4=5 > 4

. . . . . .

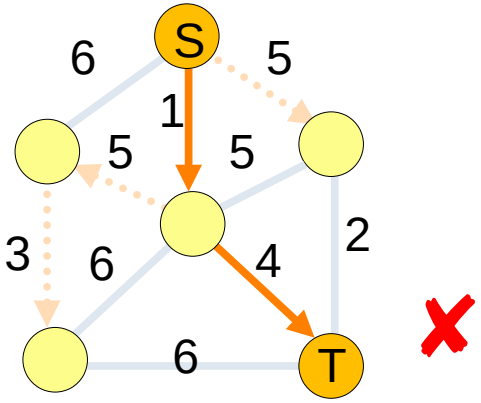
No-Instance has no *certificate*, ~~proof of yes~~

Has Spanning Tree w/ Cost <=**14** ?



1+5+3+4+2=15 > 14

Has S-T path w/ Cost <=**4** ?

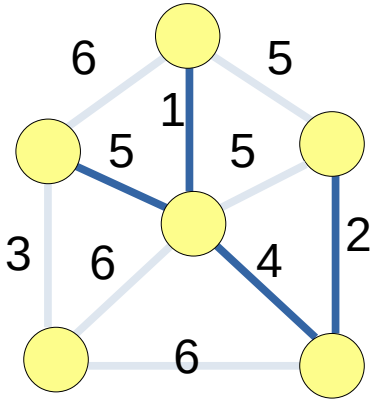


1+4=5 > 4

. . . . . .

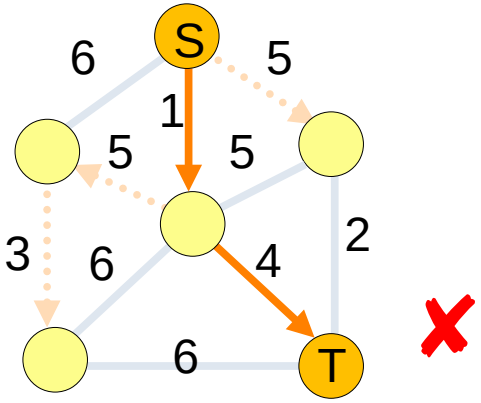
No-Instance has no *certificate*, ~~proof of yes~~

Has Spanning Tree w/ Cost <=**14** ?



1+5+4+2=12 <= 14

Has S-T path w/ Cost <=**4** ?

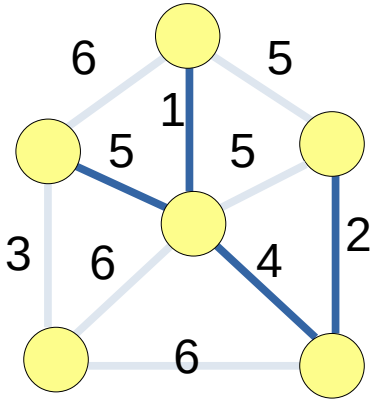


1+4=5 > 4

. . . . . .

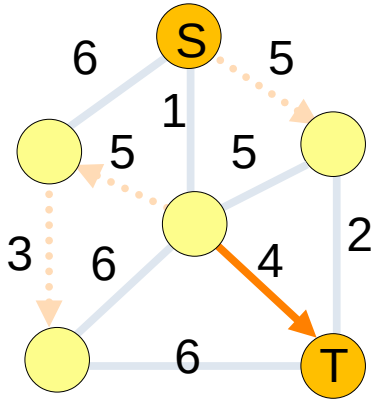
No-Instance has no *certificate*, ~~proof of yes~~

Has Spanning Tree w/ Cost <=**14** ?



1+5+4+2=12 <= 14

Has S-T path w/ Cost <=**4** ?

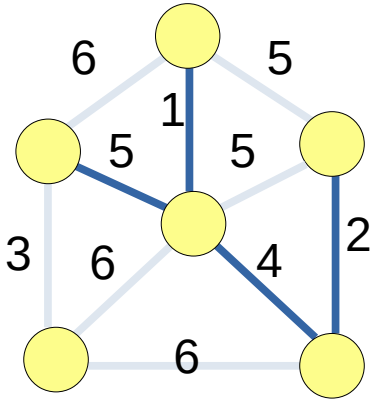


4=4 <= 4

. . . . . .

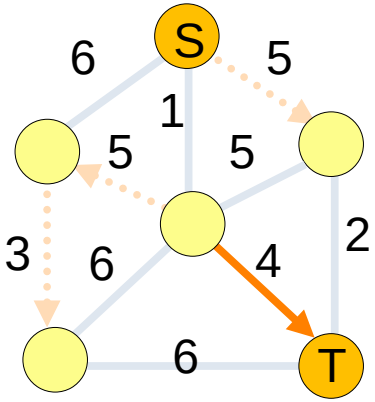
No-Instance has no *certificate*, ~~proof of yes~~

Has Spanning Tree w/ Cost <=**14** ?



1+5+4+2=12 <= 14

Has S-T path w/ Cost <=**4** ?

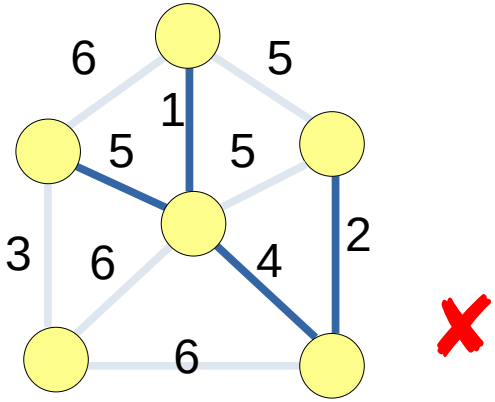


4=4 <= 4

. . . . . .

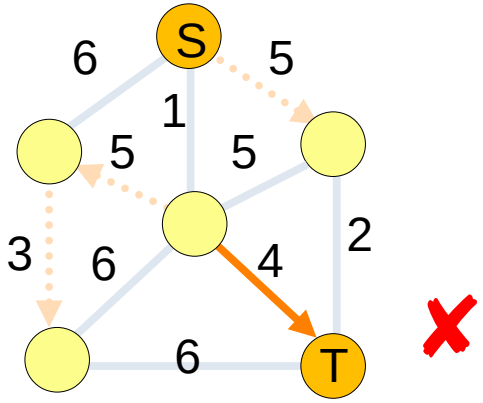
No-Instance has no *certificate*, ~~proof of yes~~

Has Spanning Tree w/ Cost <=**14** ?



1+5+4+2=12 <= 14

Has S-T path w/ Cost <=**4** ?

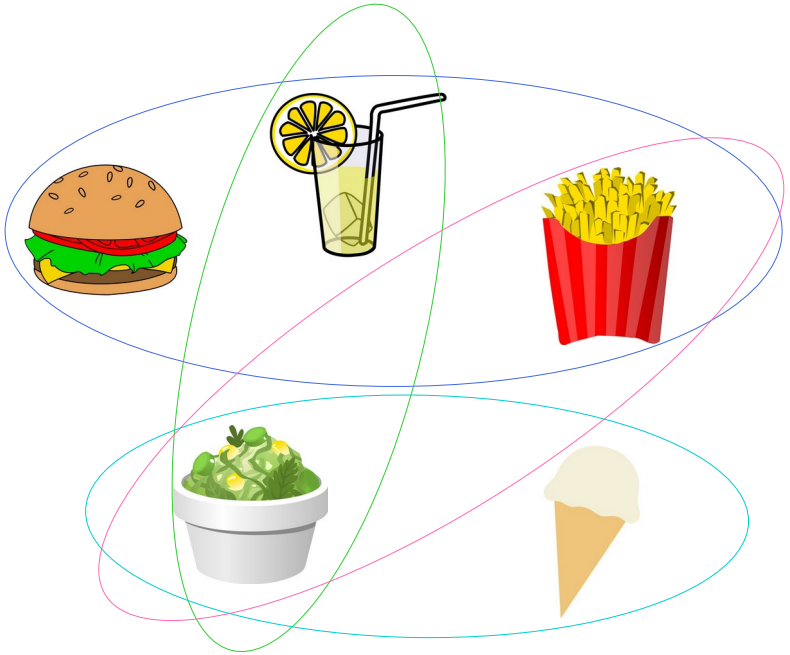


4=4 <= 4

. . . . . .

There are also many other problems...

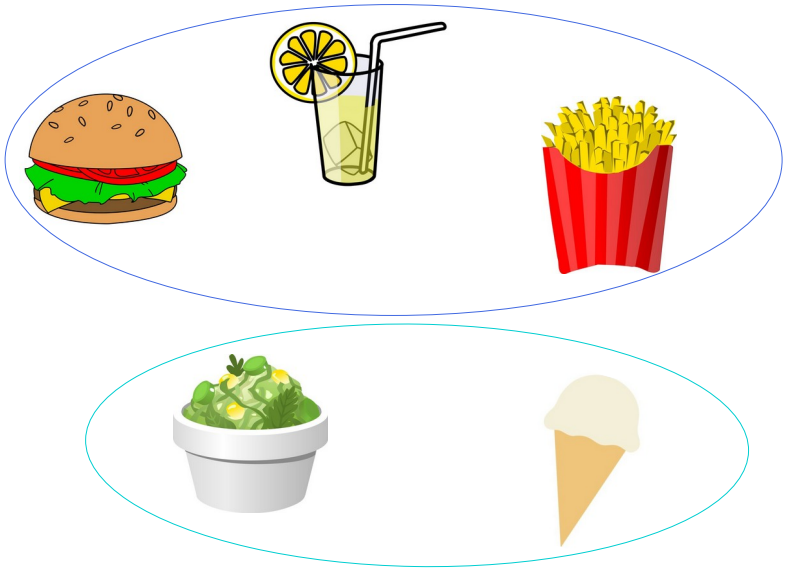
Can we get all by buying only **2** bundles?



SET-COVER

There are also many other problems...

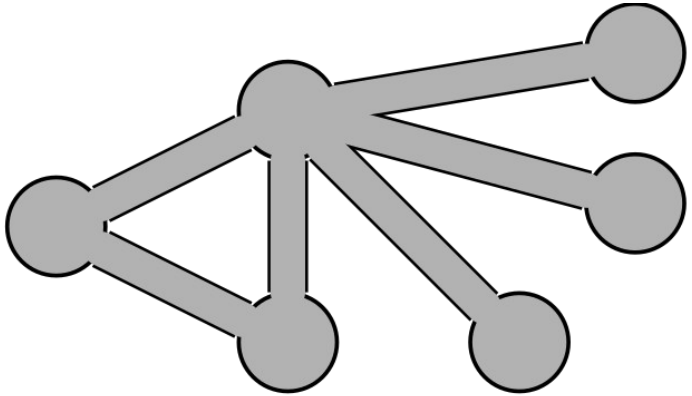
Can we get all by buying only **2** bundles?



SET-COVER

There are also many other problems...

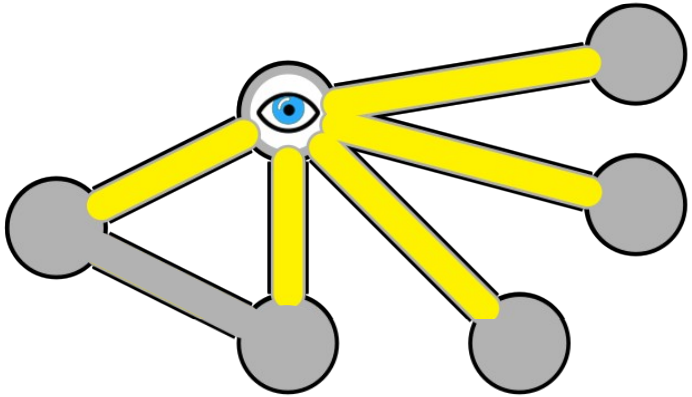
Can we watch all roads by setting only **2** sentry points?



VERTEX-COVER

There are also many other problems...

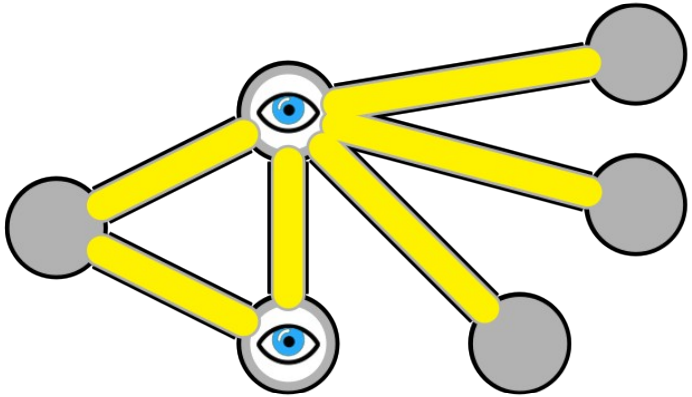
Can we watch all roads by setting only **2** sentry points?



VERTEX-COVER

There are also many other problems...

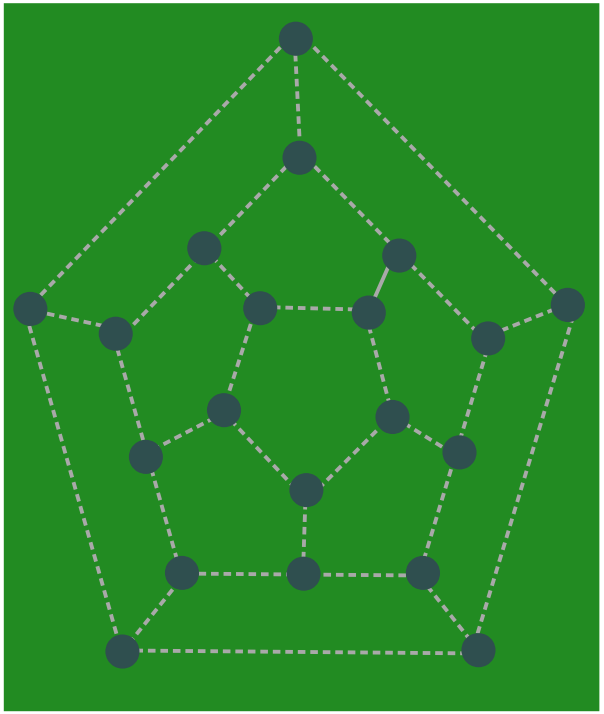
Can we watch all roads by setting only **2** sentry points?



VERTEX-COVER

There are also many other problems...

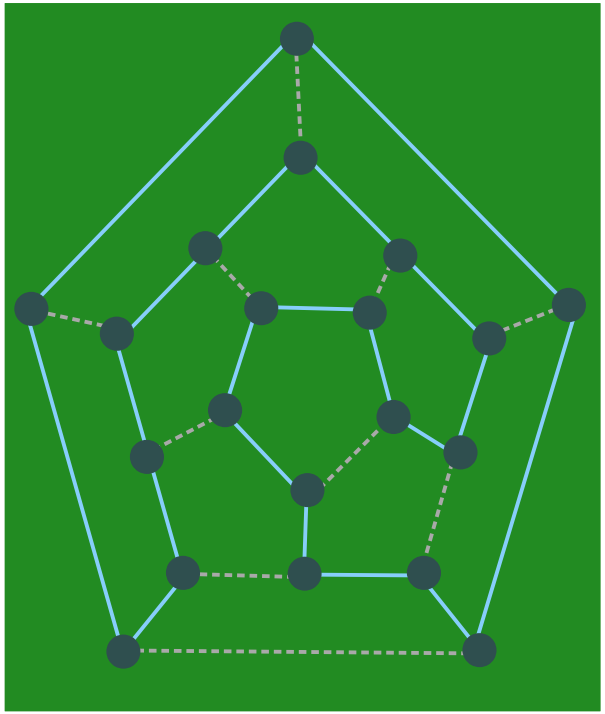
Is there a cycle that visits all vertices?



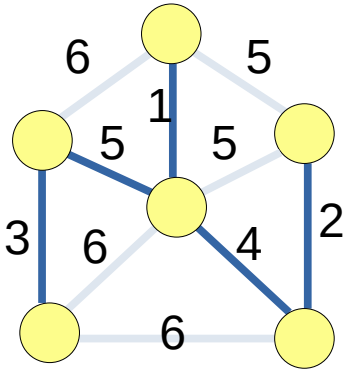
HAMILTONIAN-CYCLE

There are also many other problems...

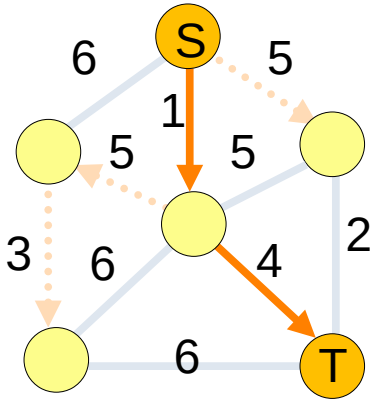
Is there a cycle that visits all vertices?



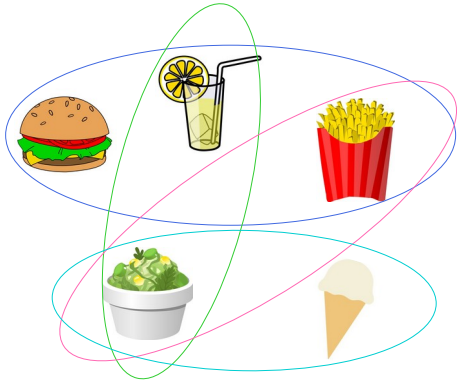
HAMILTONIAN-CYCLE



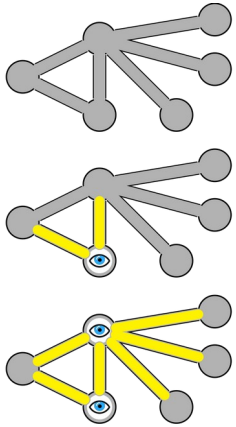
Minimum-Spanning-Tree



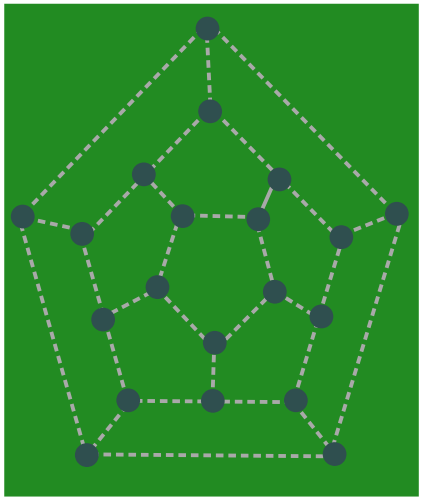
Shortest-Path



Set-Cover

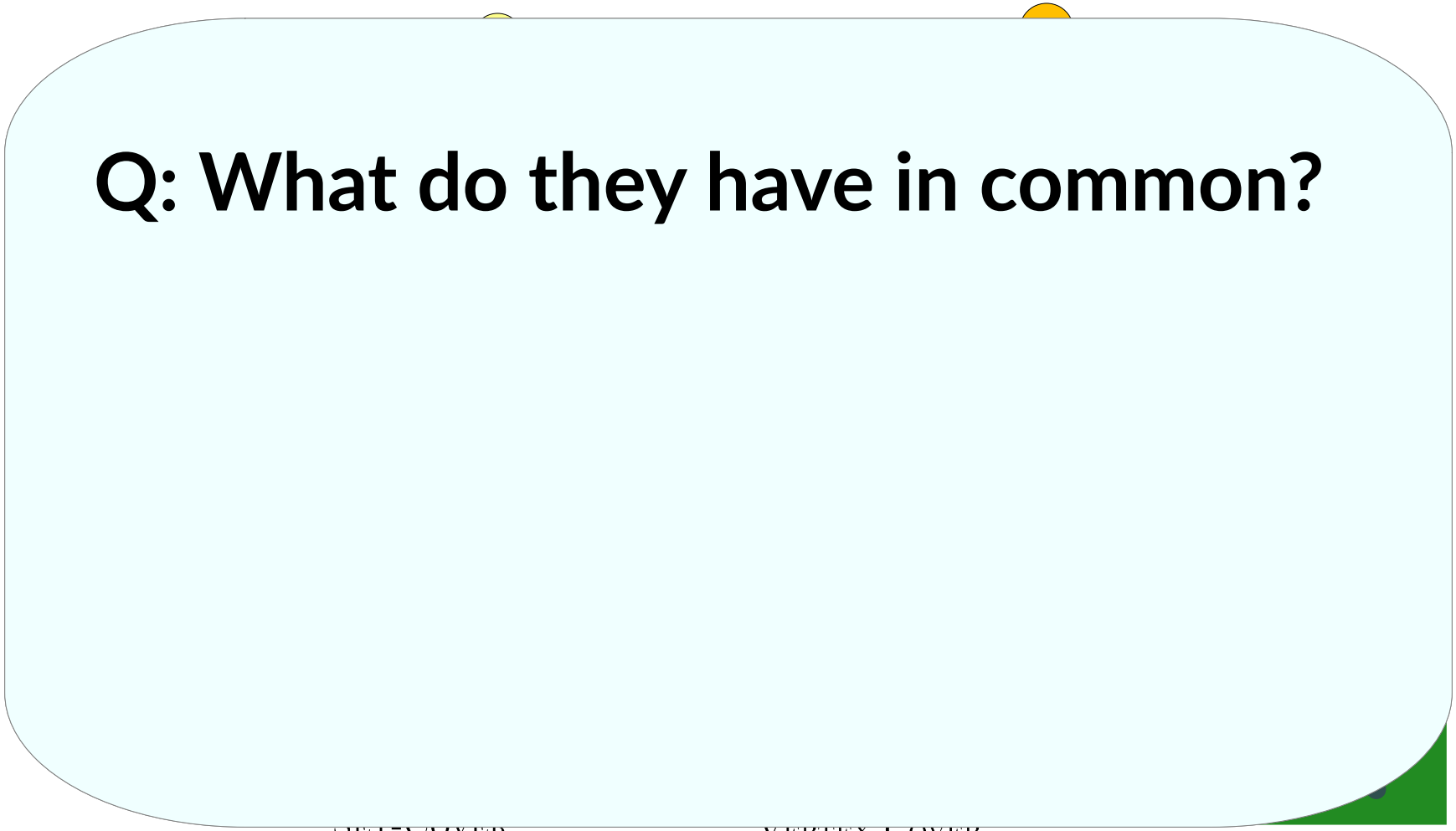


Vertex-Cover



Hamiltonian-Cycle

Q: What do they have in common?



Q: What do they have in common?

A: Validity of certificate EASY to check!
(can be done in polynomial-time)

SET-COVER

VERTEX-COVER

HAMILTONIAN-CYCLE

Q: What do they have in common?

A: Validity of certificate EASY to check!
(can be done in polynomial-time)

$O(n)$ $O(n^2)$

SET-COVER VERTEX-COVER HAMILTONIAN-CYCLE

Q: What do they have in common?

A: Validity of certificate EASY to check! (can be done in polynomial-time)

$$O(n) \quad O(n^2) \quad O(n^{10^{10}})$$

SET-COVER VERTEX-COVER HAMILTONIAN-CYCLE

Q: What do they have in common?

A: Validity of certificate EASY to check! (can be done in polynomial-time)

$$O(n) \quad O(n^2) \quad O(n^{10^{10}}) \quad \cancel{O(1.01^n)}$$

SET-COVER VERTEX-COVER HAMILTONIAN-CYCLE

Q: What do they have in common?

A: Validity of certificate EASY to check! (can be done in polynomial-time)

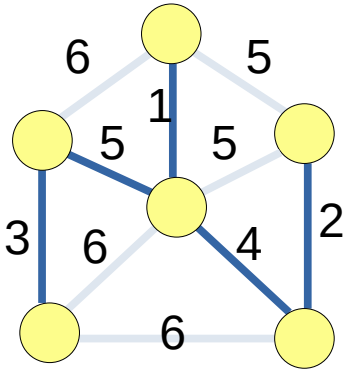
NP-Problems

(**N**on-deterministic **P**olynomial-time)

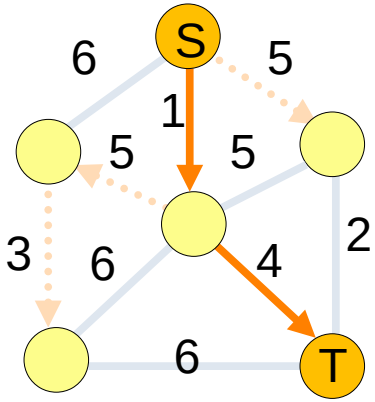
SET-COVER

VERTEX-COVER

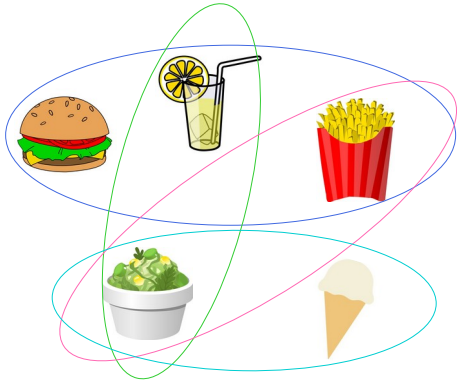
HAMILTONIAN-CYCLE



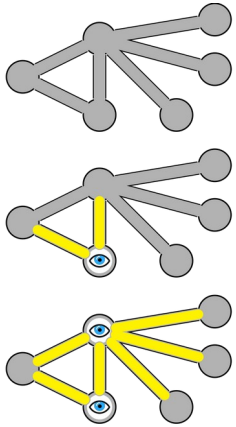
Minimum-Spanning-Tree



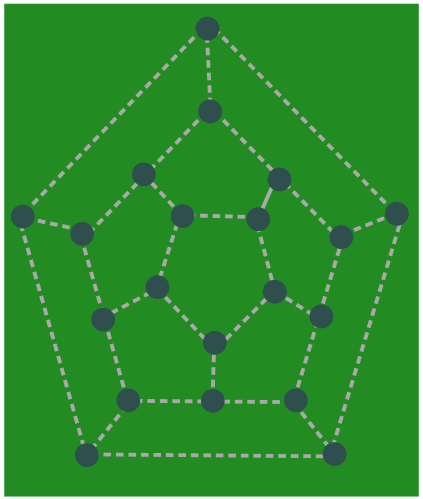
Shortest-Path



Set-Cover



Vertex-Cover



Hamiltonian-Cycle

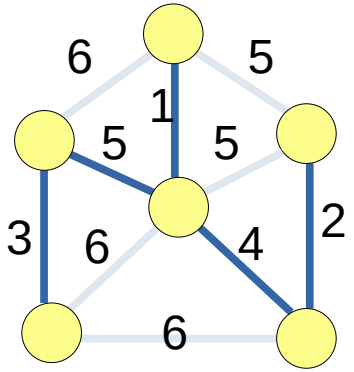
Q: Any difference?

SET-COVER

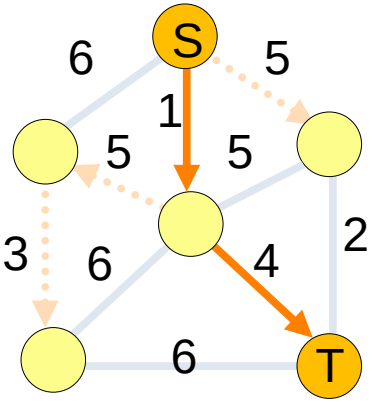
VERTEX-COVER

HAMILTONIAN-CYCLE

“Easy”

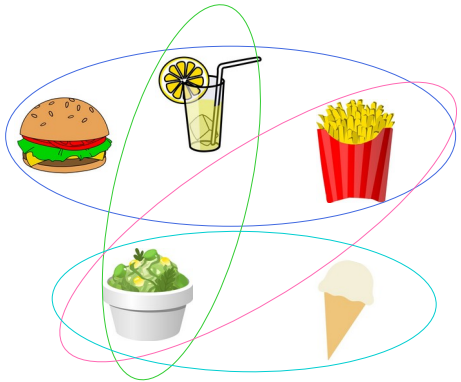


MINIMUM-SPANNING-TREE

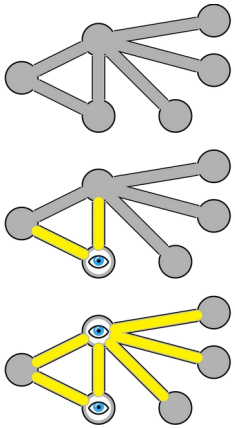


SHORTEST-PATH

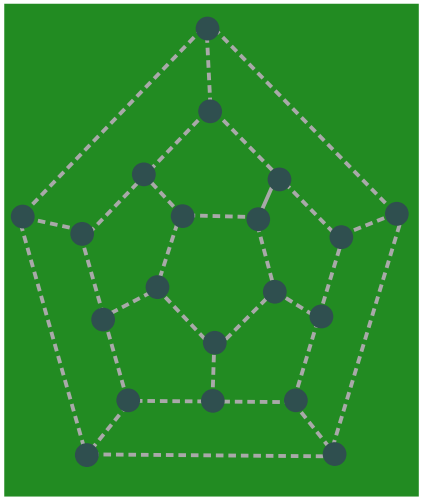
“Hard”



SET-COVER



VERTEX-COVER



HAMILTONIAN-CYCLE

Q: Any difference?

A: It is generally believed that:

"Hard" problems have NO efficient algorithms

SET-COVER

VERTEX-COVER

HAMILTONIAN-CYCLE

Q: Any difference?

A: It is generally believed that:

"Hard" problems have NO efficient algorithms

But there's no proof for it yet...

SET-COVER

VERTEX-COVER

HAMILTONIAN-CYCLE

Q: Any difference?

A: It is generally believed that:

"Hard" problems have NO efficient algorithms

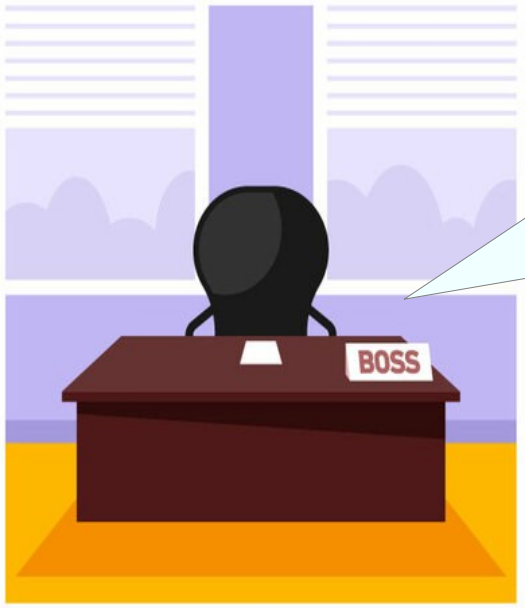
But there's no proof for it yet...

SET-COVER

VERTEX-COVER

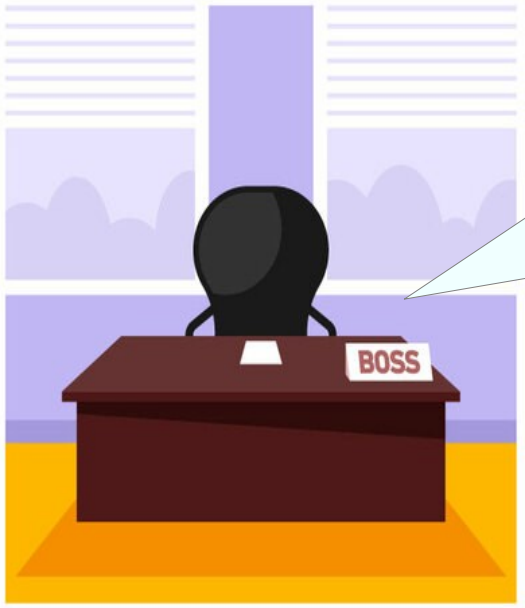
HAMILTONIAN-CYCLE

How do you prove that an NP-problem is “Hard”?



Design an efficient algorithm for problem N!

How do you prove that an NP-problem is “Hard”?



Design an efficient algorithm for problem N!

But… problem N is “Hard”

How do you prove that an NP-problem is “Hard”?

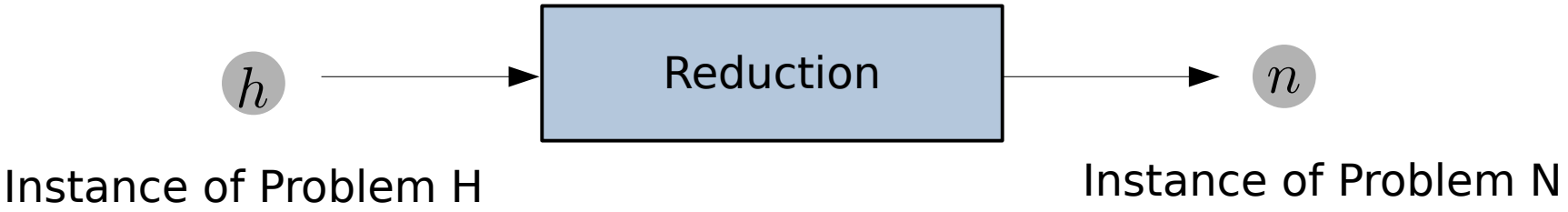
If N could be solved,
a known hard problem H
could be also solved.

How do you prove that an NP-problem is “Hard”?

“reduction”

If N could be solved,
a known hard problem H
could be also solved.

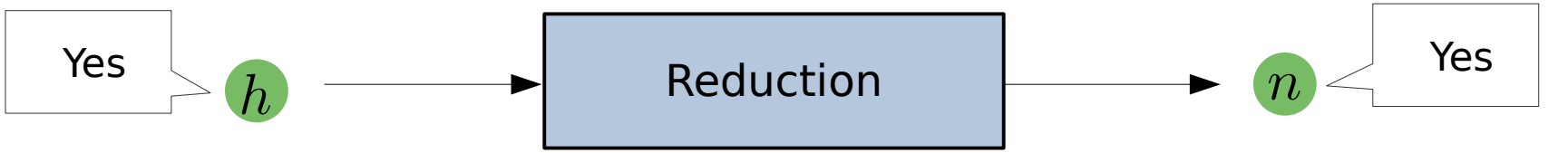
One-Call Reduction



One-Call Reduction – Correctness Property

H is the problem known to be hard

n is the new problem



Instance of Problem H

Instance of Problem N

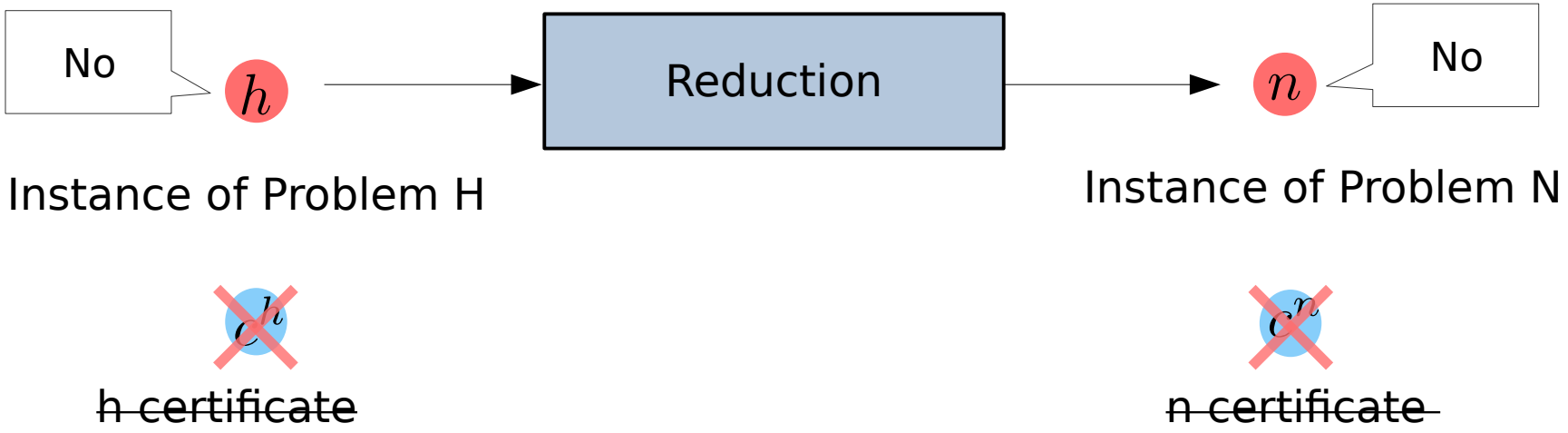
$\exists$ c^h

h certificate

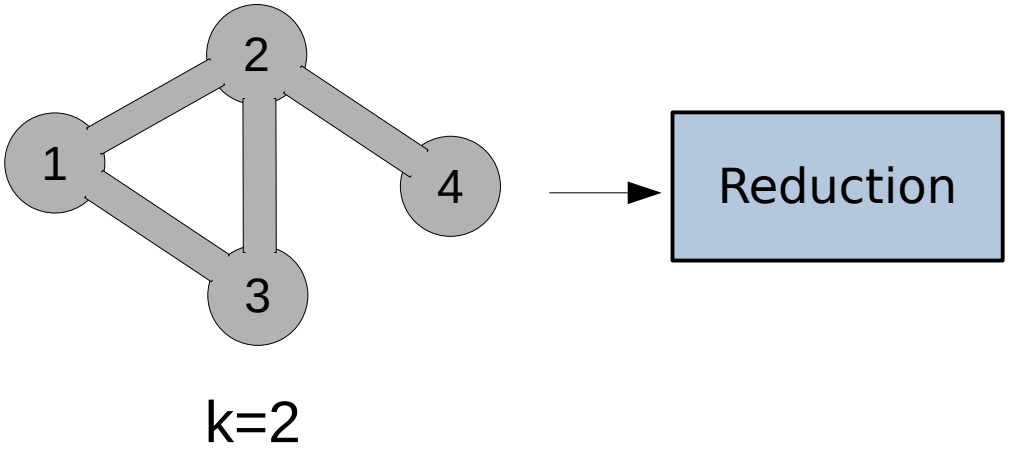
$\exists$ c^n

n certificate

One-Call Reduction – Correctness Property



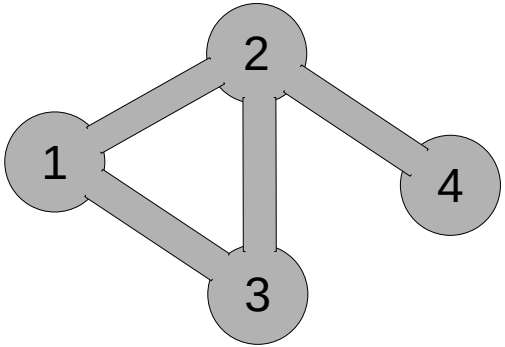
One-Call Reduction



VERTEX-COVER

SET-COVER

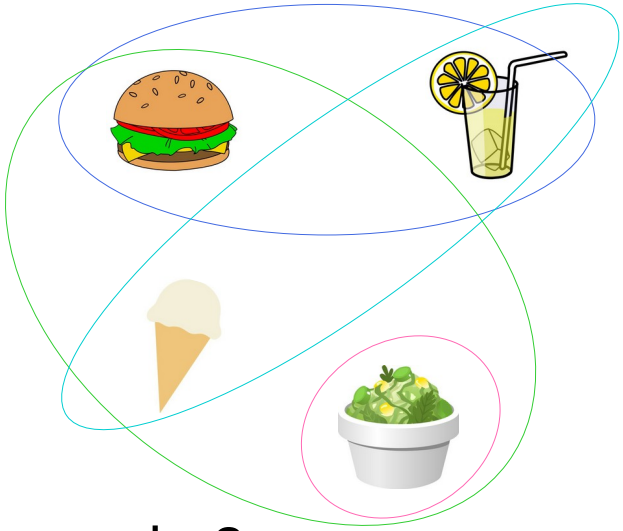
One-Call Reduction



k=2

VERTEX-COVER

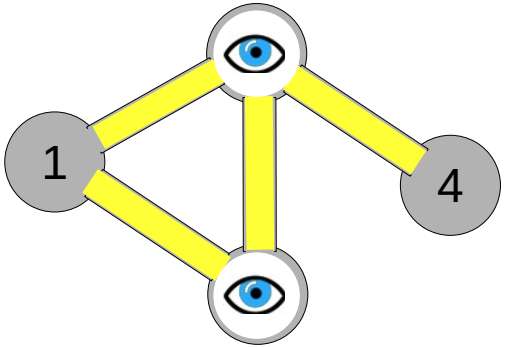
Reduction



k=2

SET-COVER

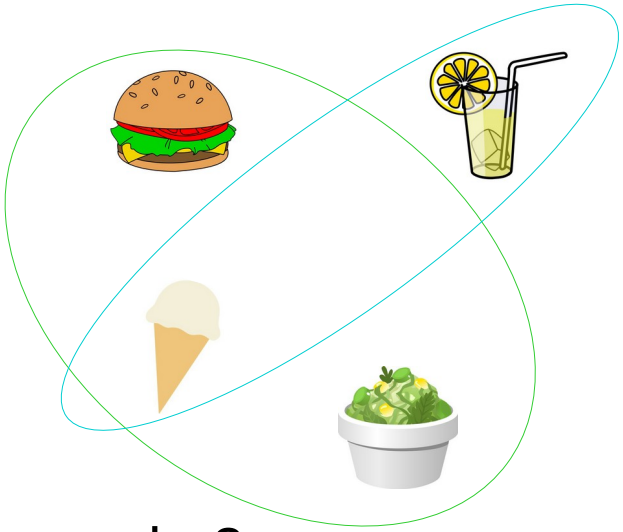
One-Call Reduction



k=2

VERTEX-COVER

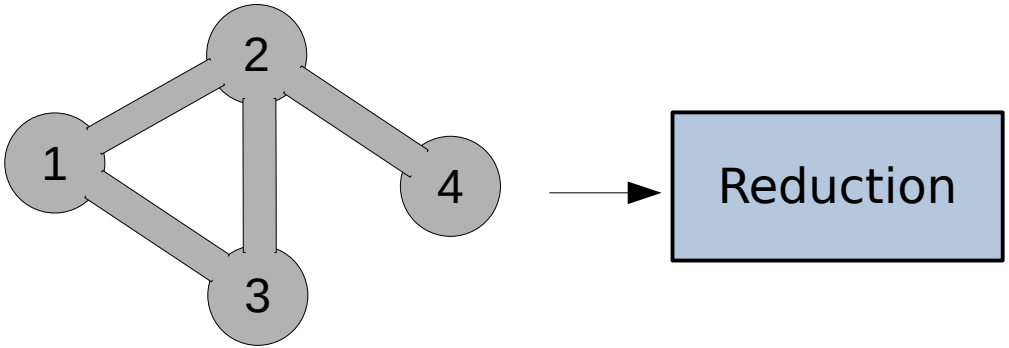
Reduction



k=2

SET-COVER

One-Call Reduction

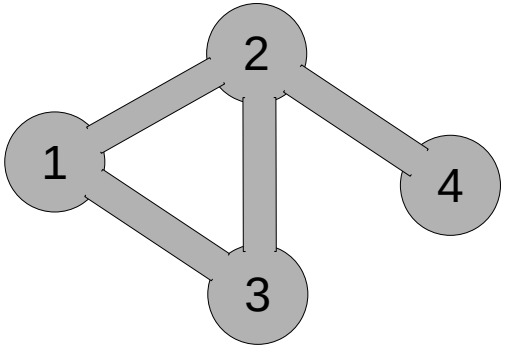


k=1

VERTEX-COVER

SET-COVER

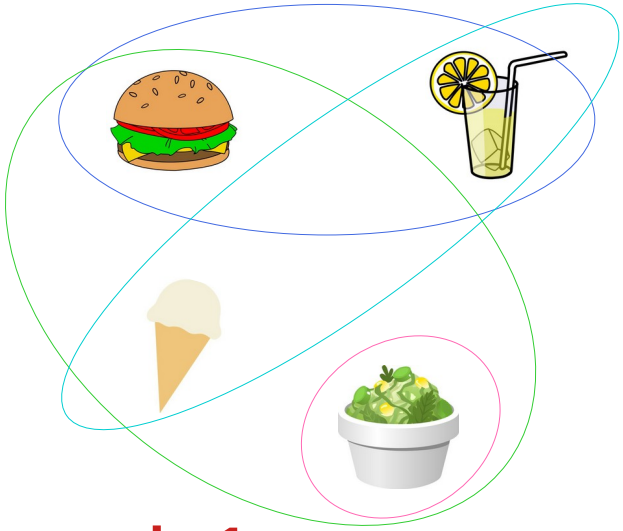
One-Call Reduction



k=1

VERTEX-COVER

Reduction



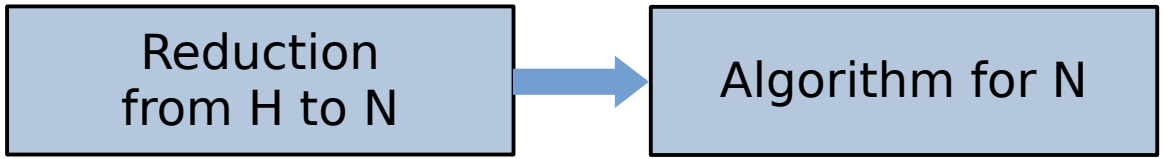
k=1

SET-COVER

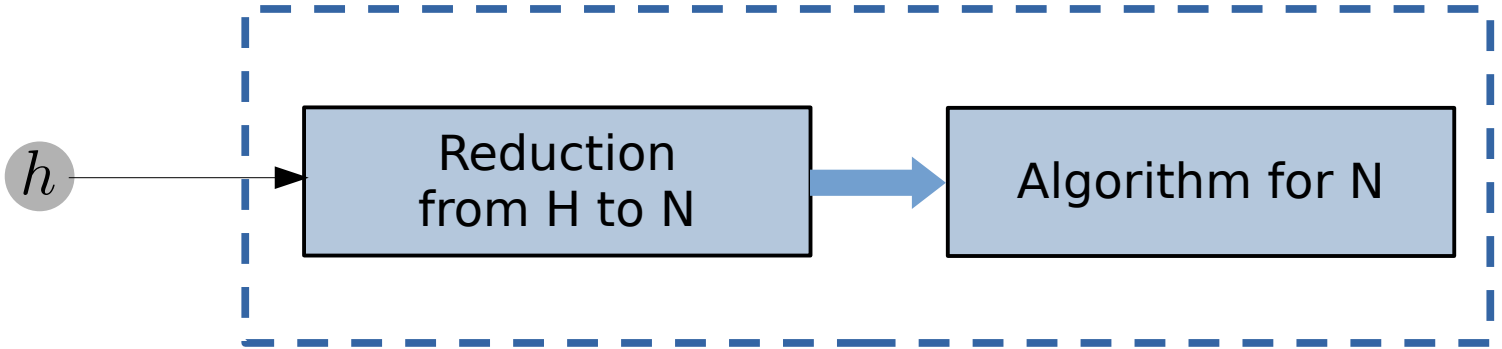
One-Call Reduction

Suppose there is an algorithm for N

One-Call Reduction

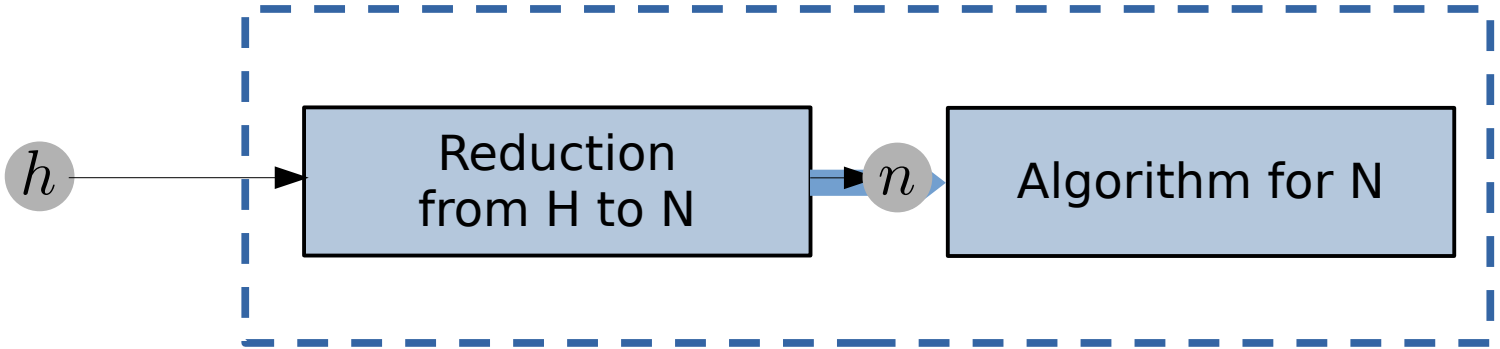


One-Call Reduction



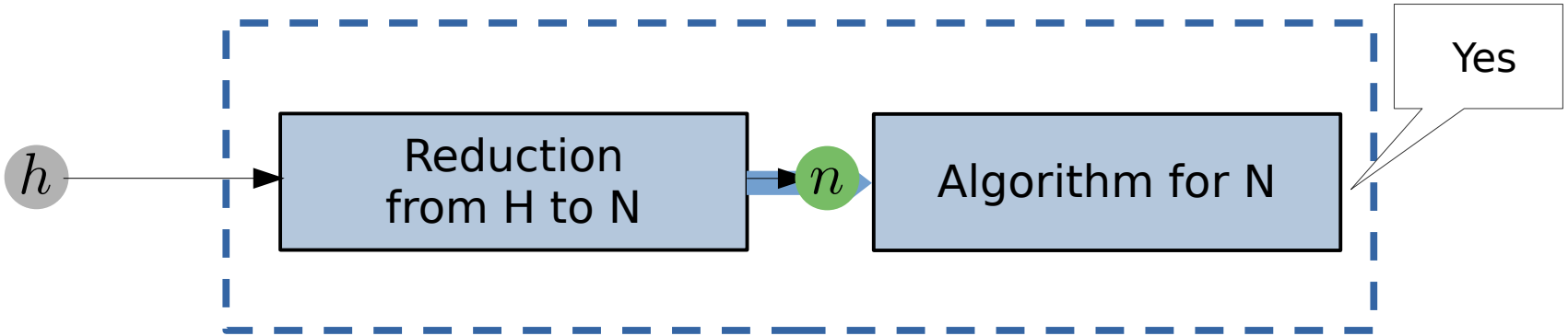
Algorithm for H

One-Call Reduction



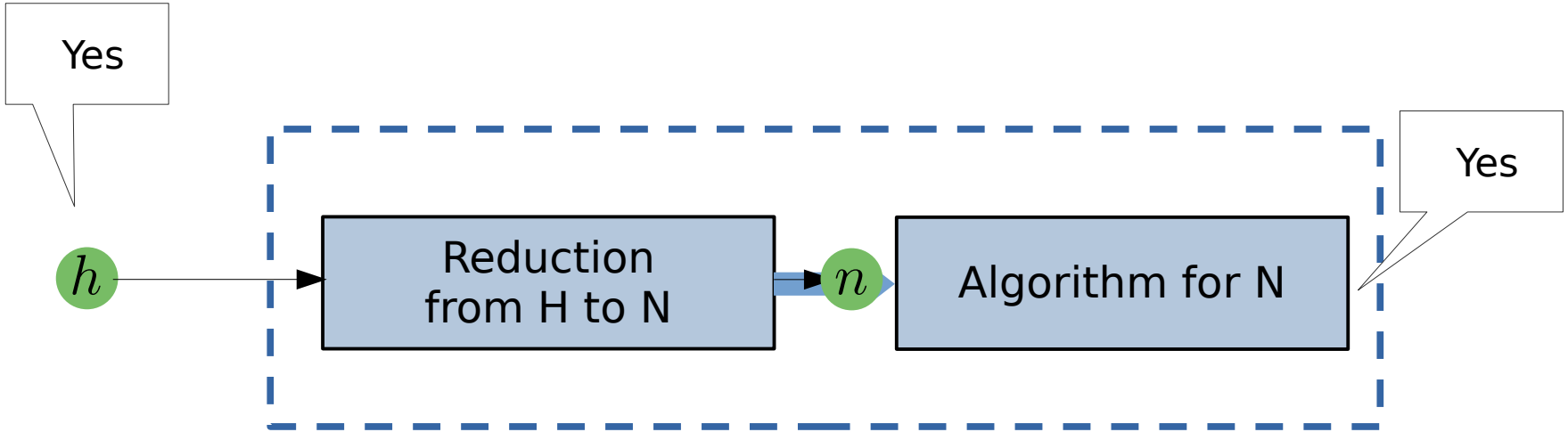
Algorithm for H

One-Call Reduction

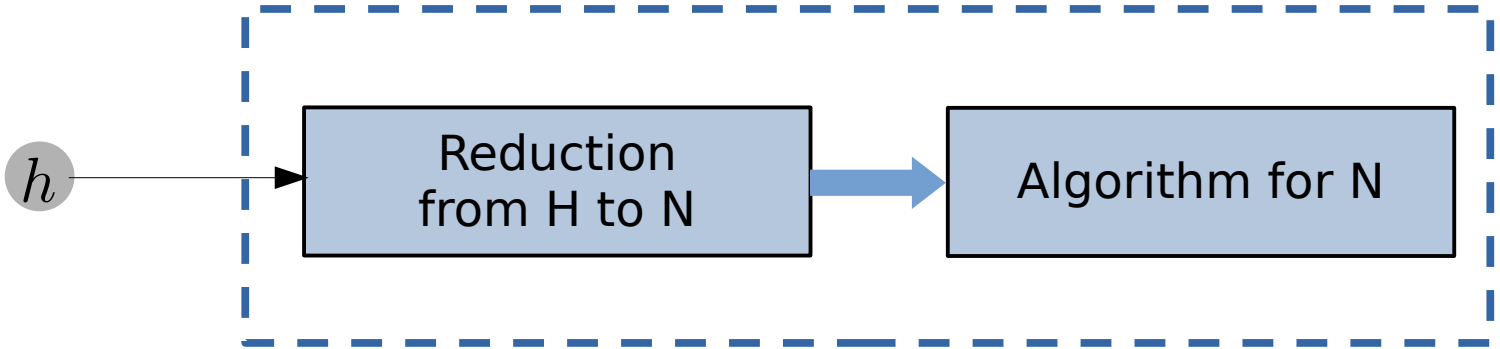


Algorithm for H

One-Call Reduction

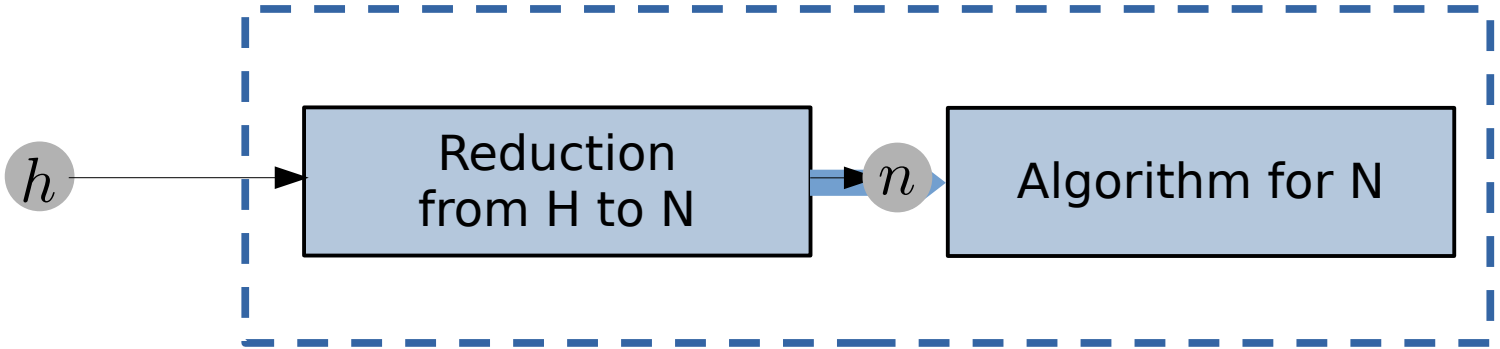


One-Call Reduction



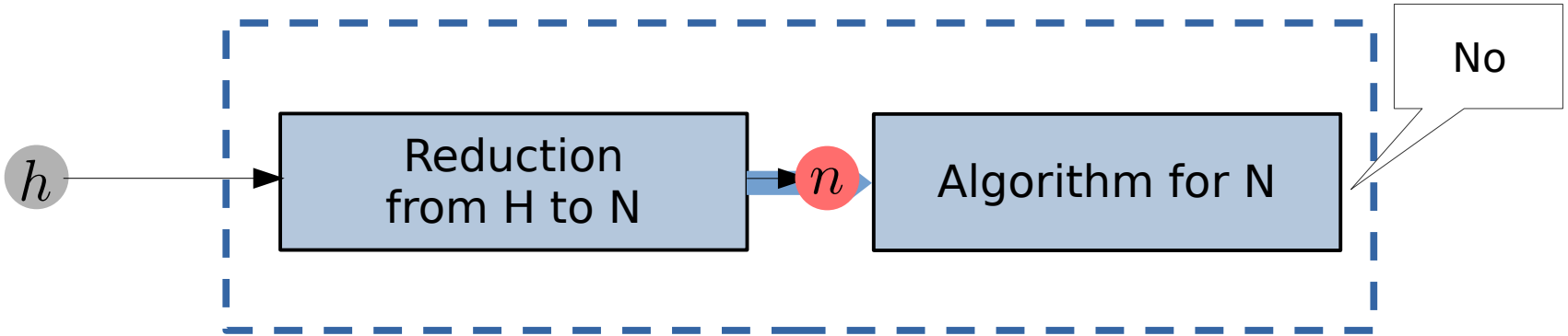
Algorithm for H

One-Call Reduction



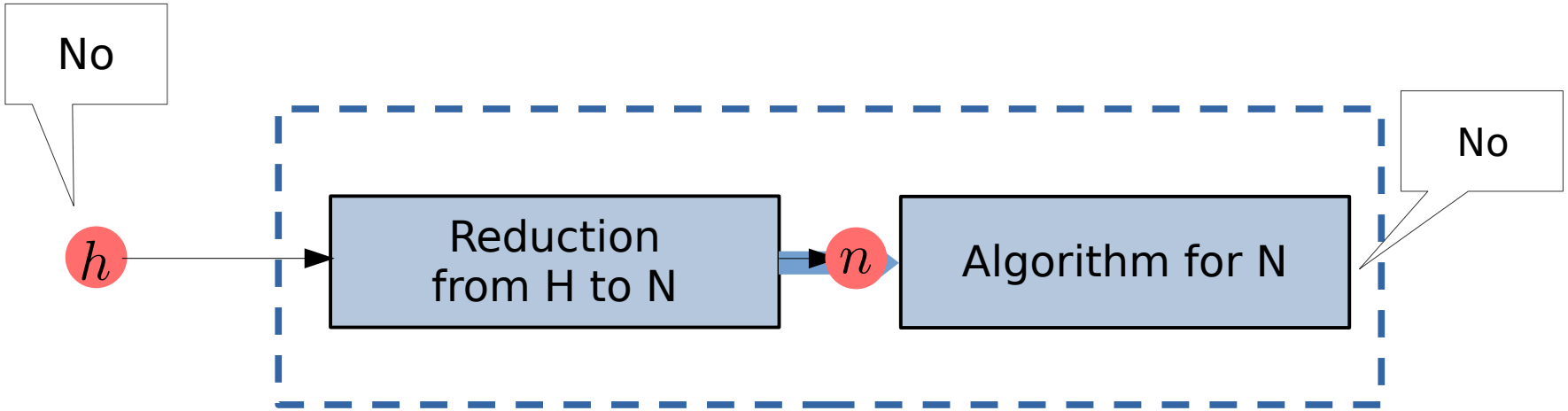
Algorithm for H

One-Call Reduction



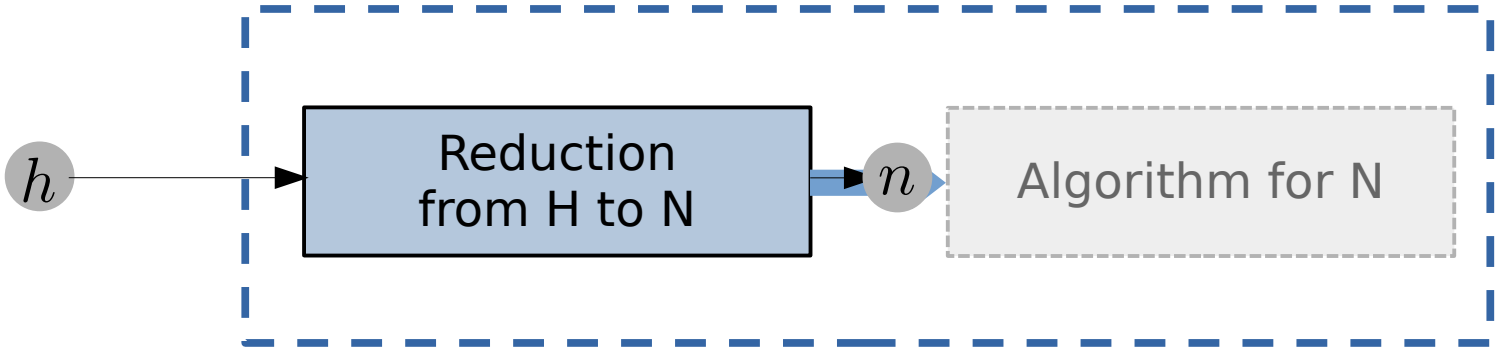
Algorithm for H

One-Call Reduction

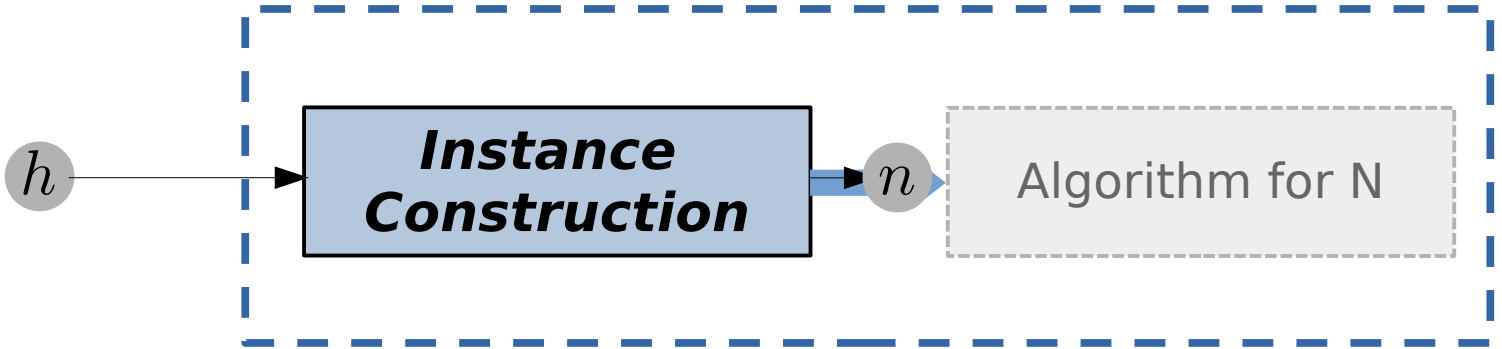


Algorithm for H

Reduction and Justifications of Correctness



Call this part "*instance construction*" from now on

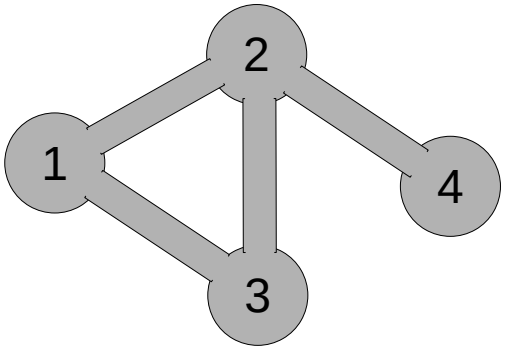


Instance Construction

VERTEX-COVER

SET-COVER

Instance Construction

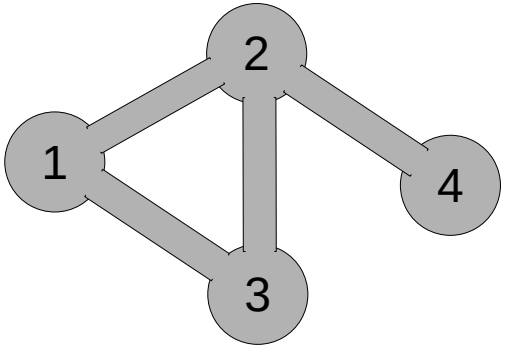


k=2

VERTEX-COVER

SET-COVER

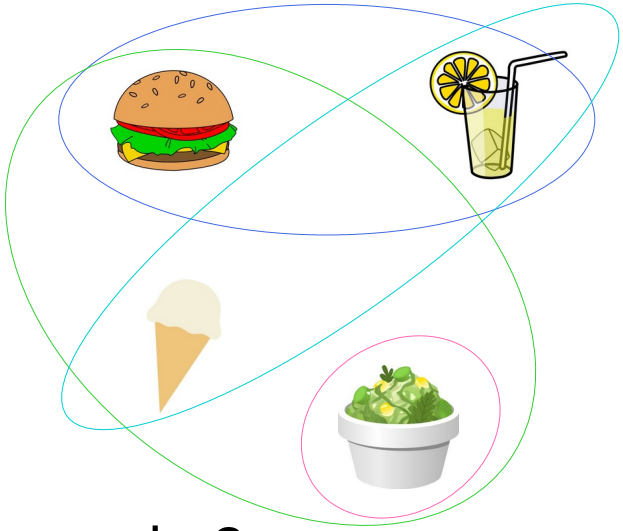
Instance Construction



k=2

VERTEX-COVER

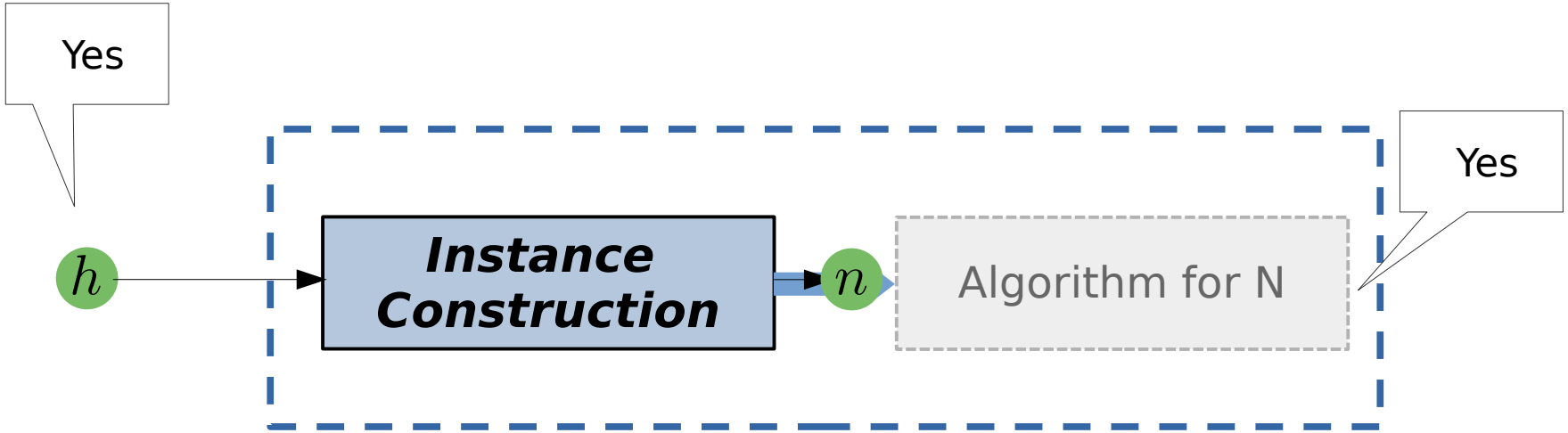
Instance Construction



k=2

SET-COVER

Justifying N Yes => H Yes



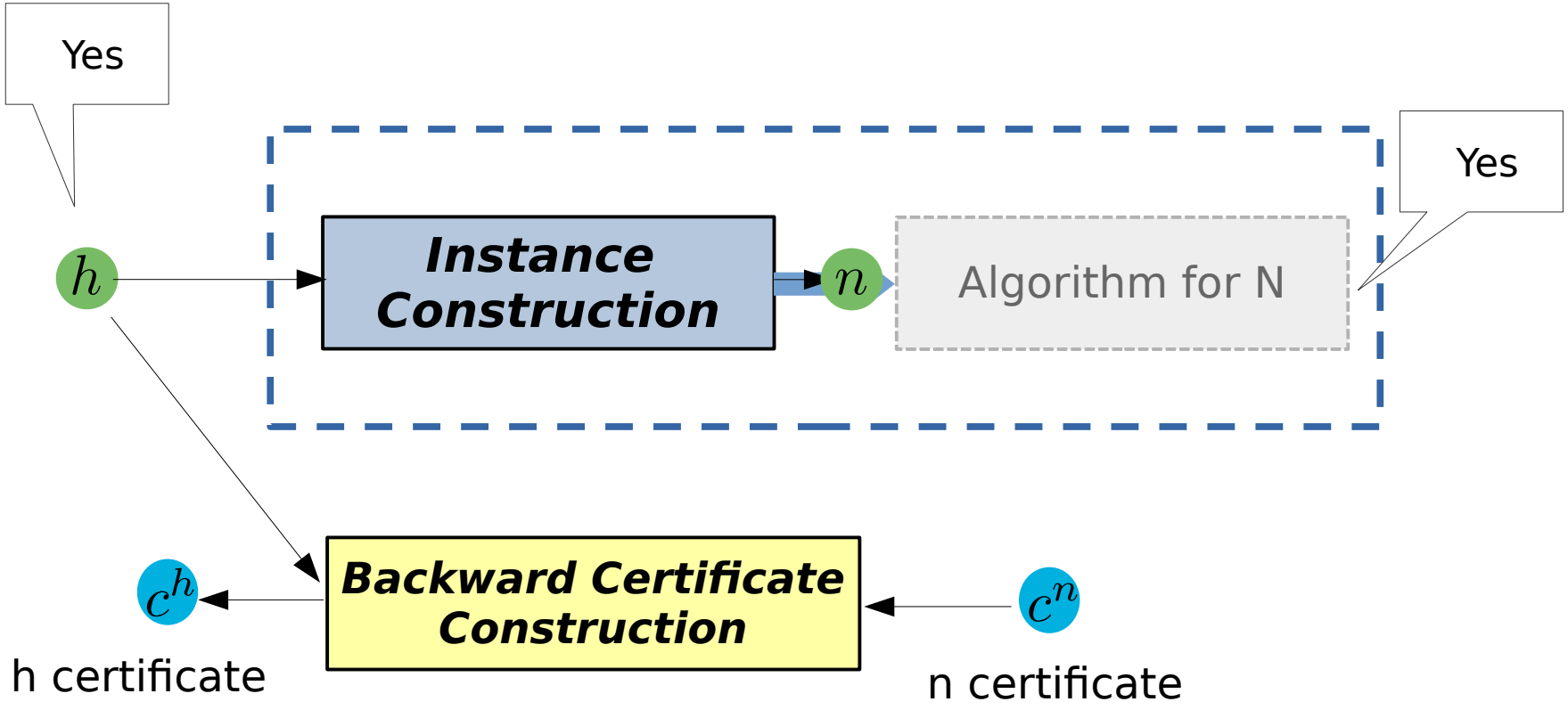
$\exists$ c^h

h certificate

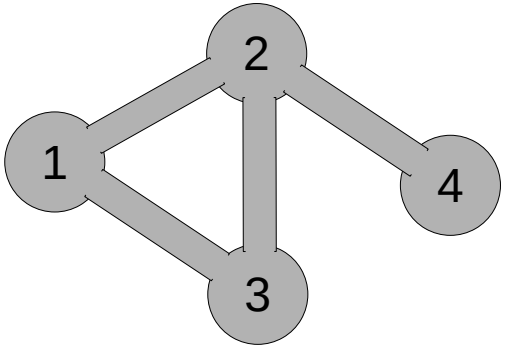
$\exists$ c^n

n certificate

Justifying N Yes => H Yes



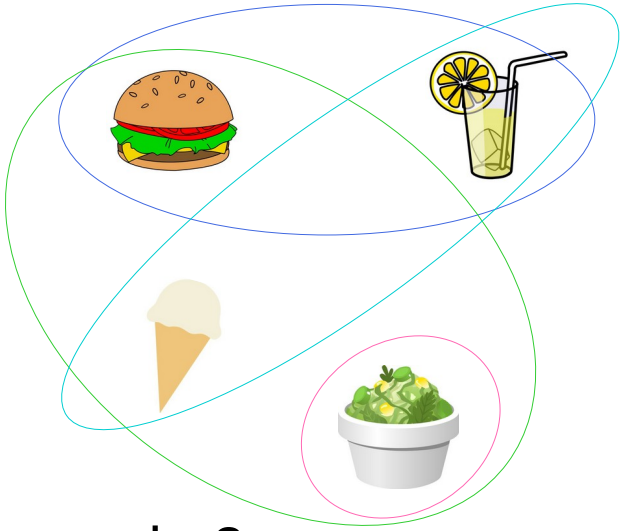
Backward Certificate Construction



k=2

VERTEX-COVER

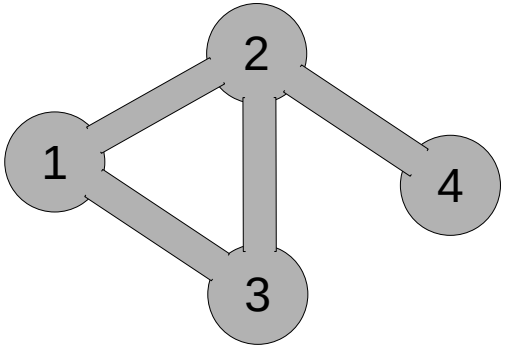
Instance Construction



k=2

SET-COVER

Backward Certificate Construction

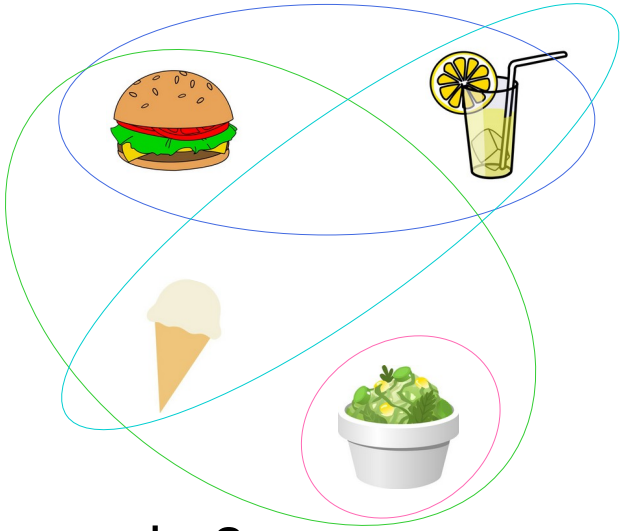


k=2

VERTEX-COVER

Instance Construction

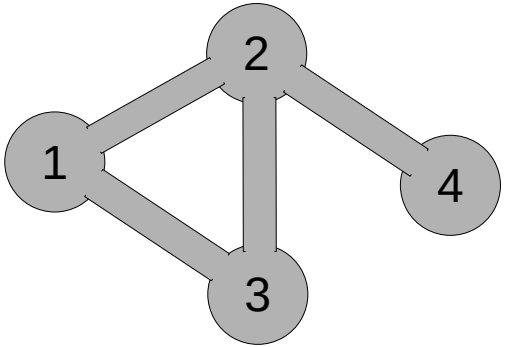
Backward Certificate Construction



k=2

SET-COVER

Backward Certificate Construction

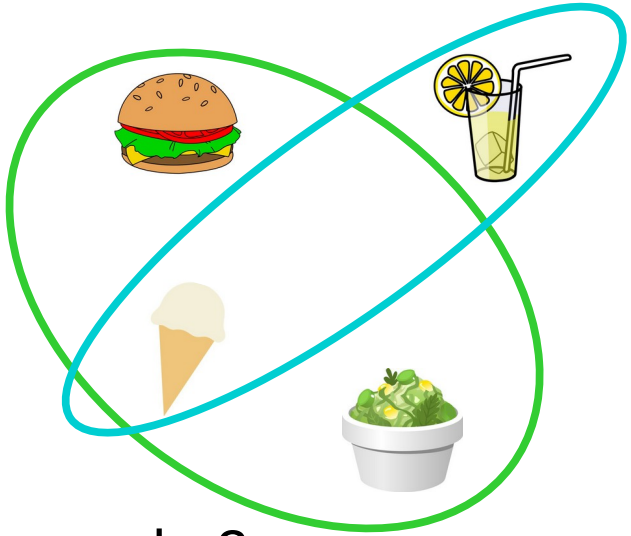


k=2

VERTEX-COVER

Instance Construction

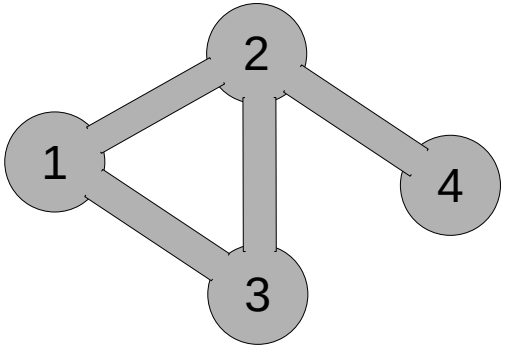
Backward Certificate Construction



k=2

SET-COVER

Backward Certificate Construction

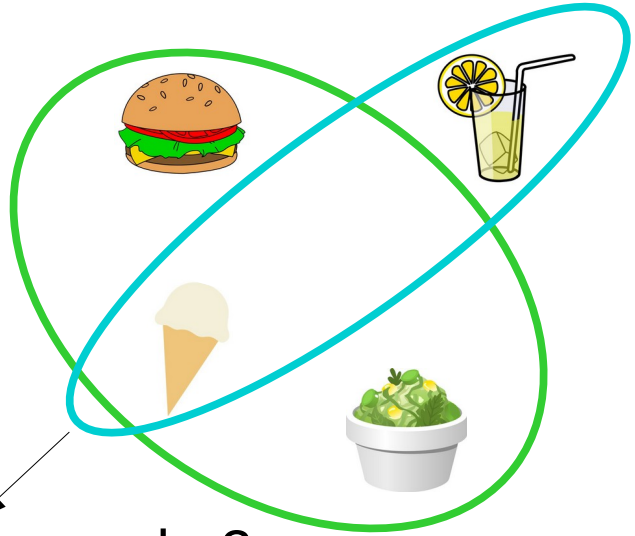


k=2

VERTEX-COVER

Instance Construction

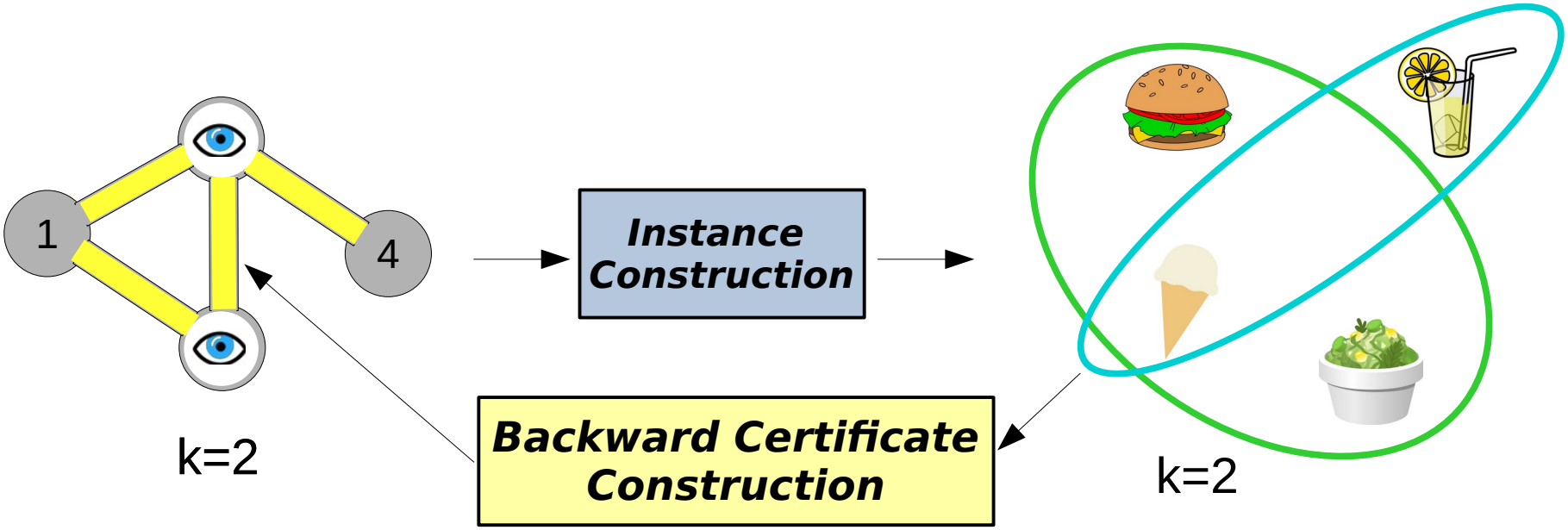
Backward Certificate Construction



k=2

SET-COVER

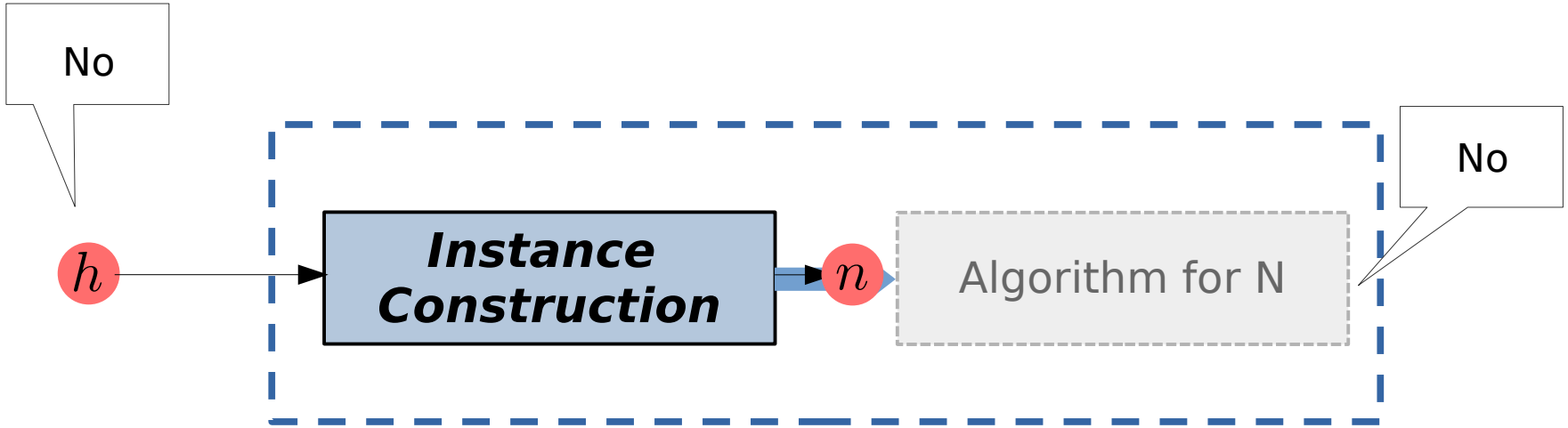
Backward Certificate Construction



VERTEX-COVER

SET-COVER

Justifying N No => H No



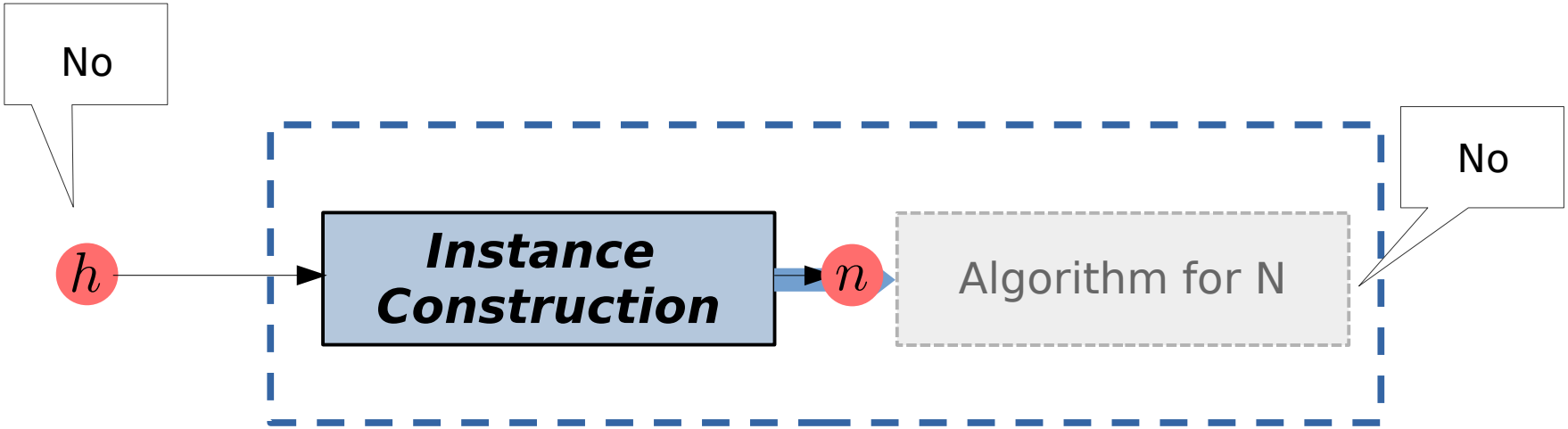
c^h

~~h certificate~~

c^n

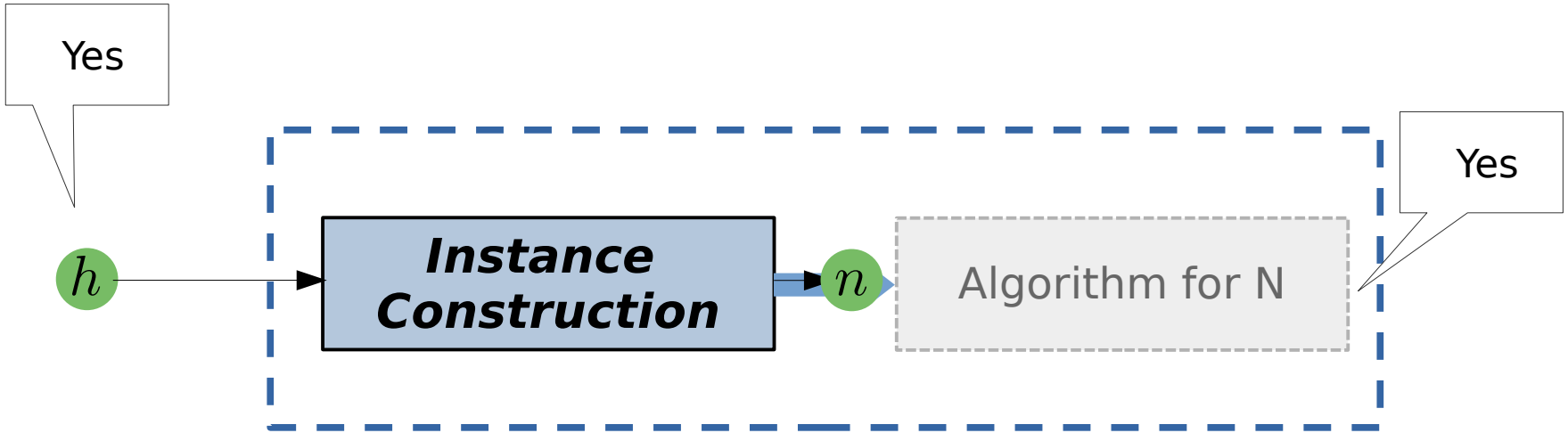
~~n certificate~~

Justifying N No => H No



$\neg\exists\ c^x \Longleftarrow \neg\exists\ c^y$

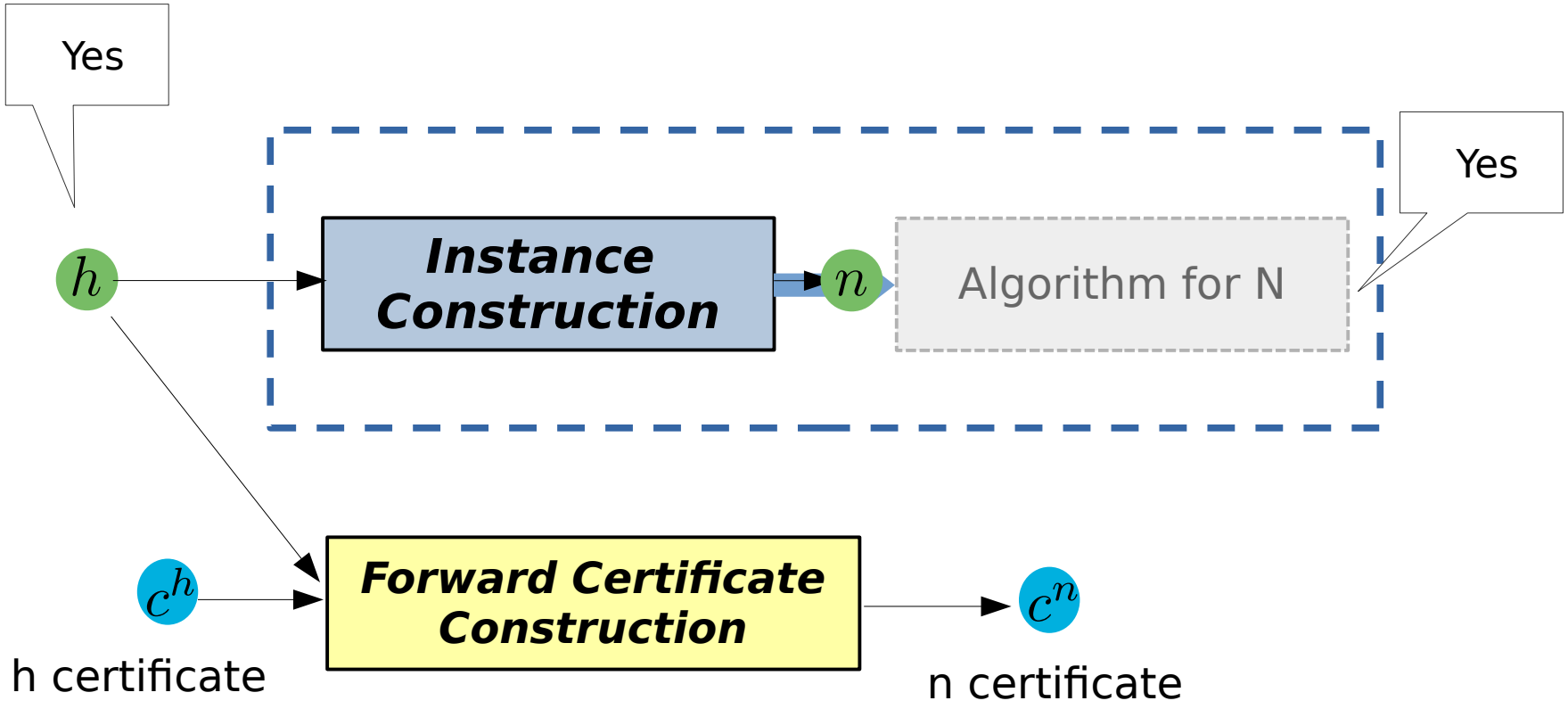
Justifying N No => H No



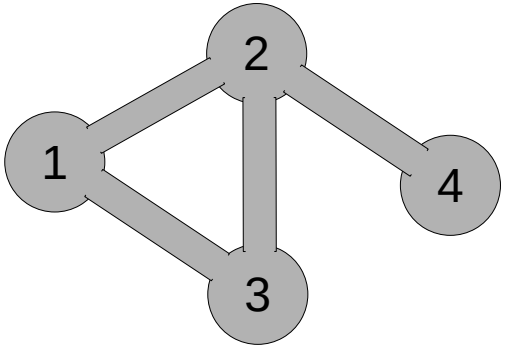
$\exists\ c^x \implies \exists\ c^y$

* we are in a classical world

Justifying N No => H No



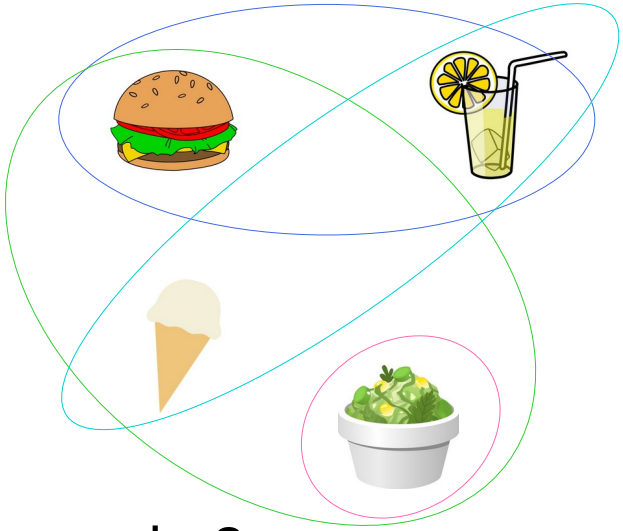
Forward Certificate Construction



k=2

VERTEX-COVER

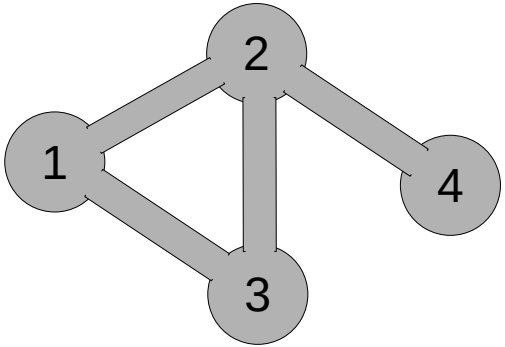
Instance Construction



k=2

SET-COVER

Forward Certificate Construction

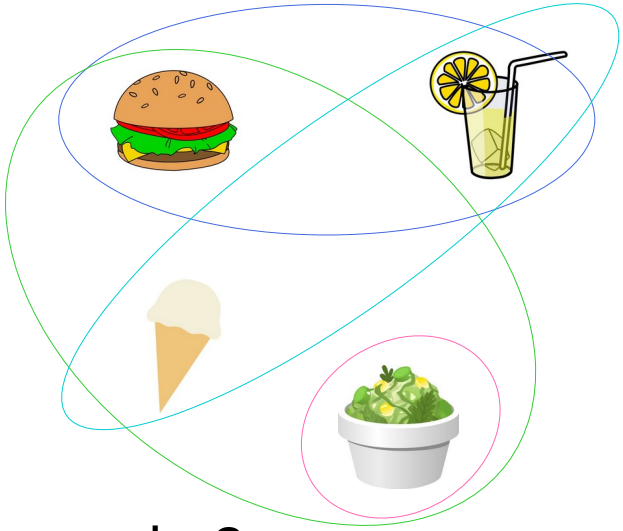


k=2

VERTEX-COVER

Instance Construction

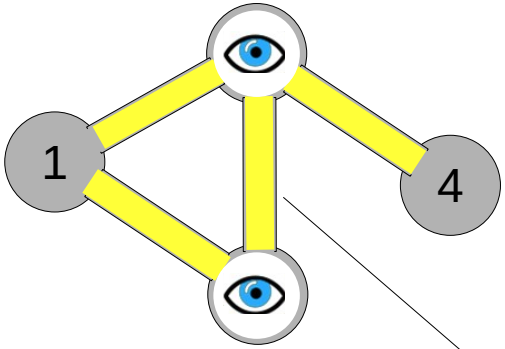
Forward Certificate Construction



k=2

SET-COVER

Forward Certificate Construction

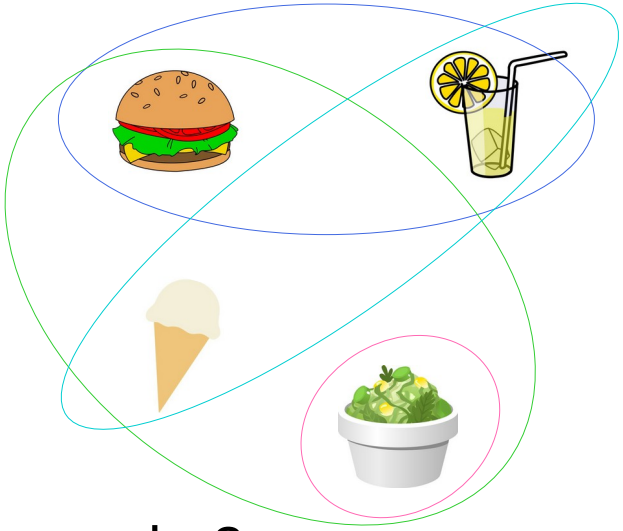


k=2

VERTEX-COVER

Instance Construction

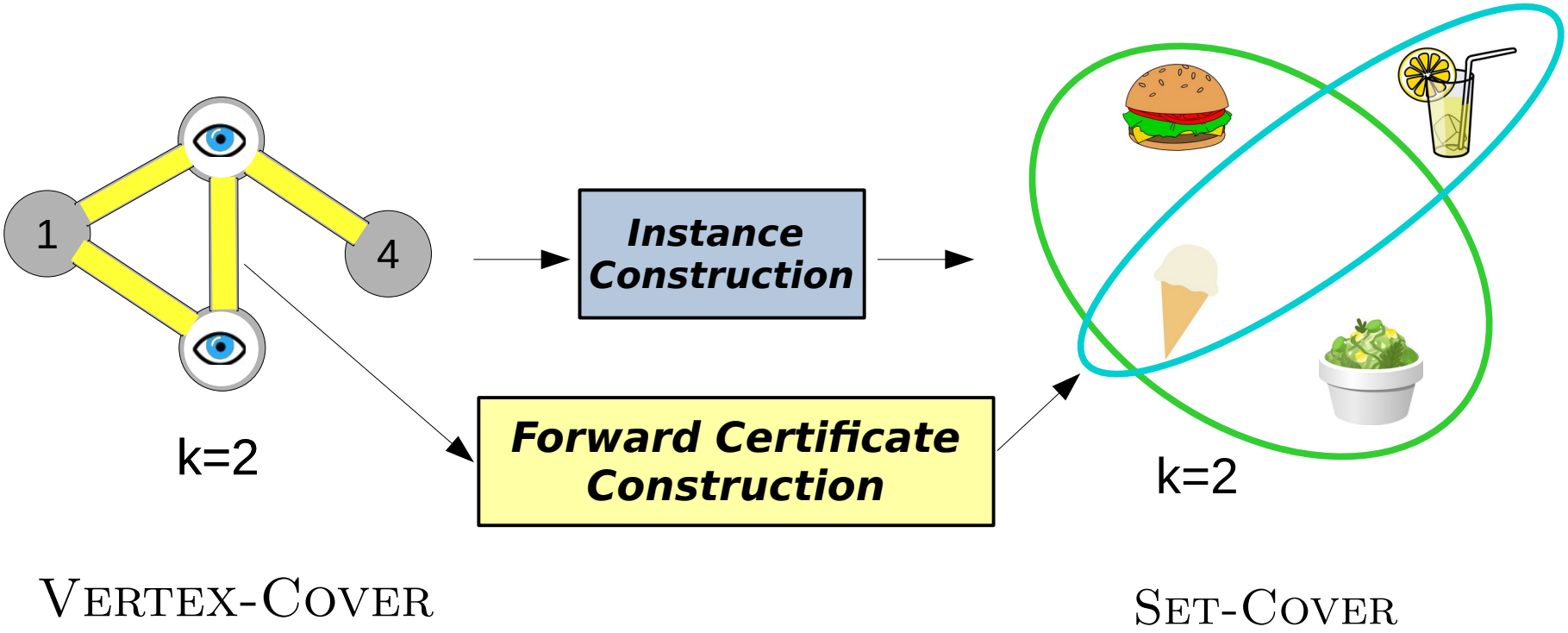
Forward Certificate Construction



k=2

SET-COVER

Forward Certificate Construction



VERTEX-COVER

More Formally

VERTEX-COVER Instance and Certificate

Instance:

Certificate:

Vertex-Cover Instance and Certificate

Instance: graph G and natural k

Certificate:

Vertex-Cover Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Vertex-Cover Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Assertion for valid certificate C of (G,k):

Vertex-Cover Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Assertion for valid certificate C of (G,k):

Forall e in edges of G:

Exists v in C s.t. v in endpoint of e

VERTEX-COVER Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Assertion for valid certificate C of (G,k):

Forall e in edges of G:

Exists v in C s.t. v in endpoint of e

and

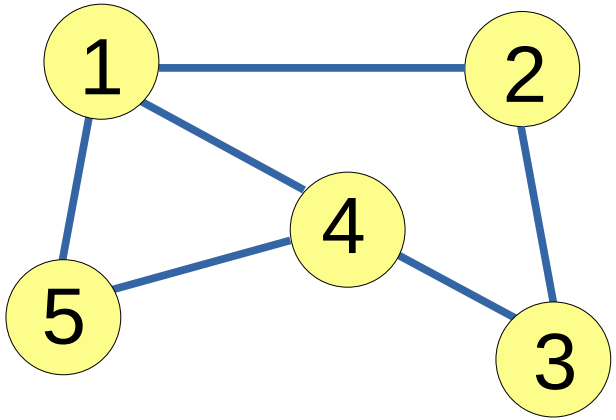
Size of C <= k

Two Important Decision Problems

(we’ll work with them on Wednesday)

Independent-Set

Exists a set of k vertices s.t. no two are neighbors of each other?

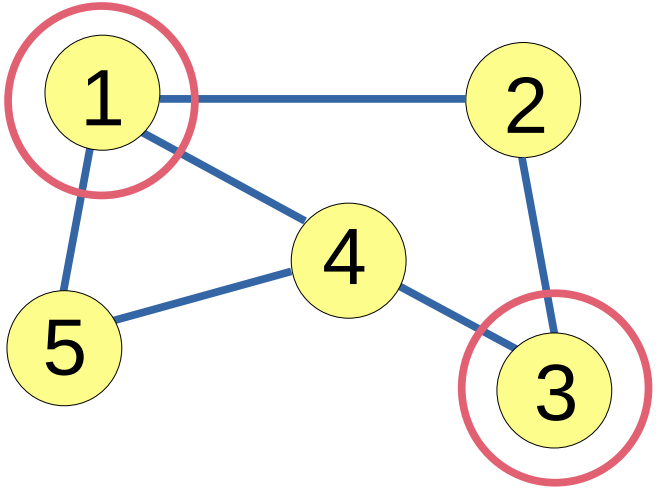


k=2

Instance: a graph G and a threshold number k

Independent-Set

Exists a set of k vertices s.t. no two are neighbors of each other?



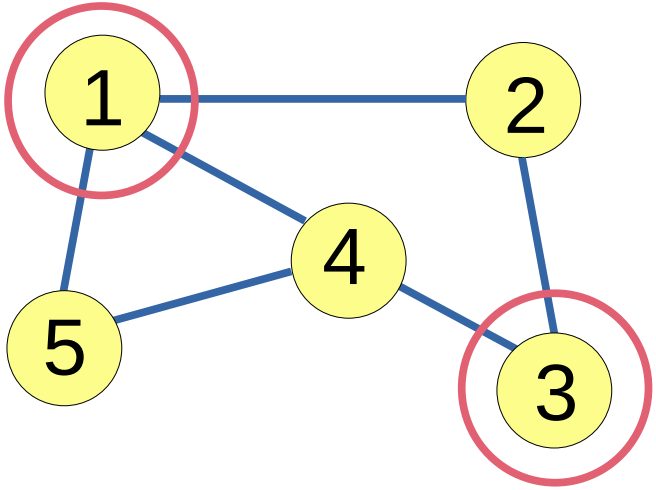
k=2

Instance: a graph G and a threshold number k

Certificate: a subset of the vertices of G

Independent-Set

Exists a set of k vertices s.t. no two are neighbors of each other?



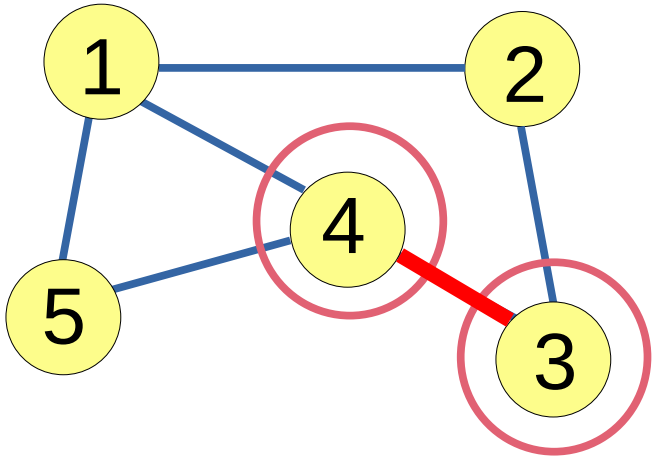
Valid Certificate

Instance: a graph G and a threshold number k

Certificate: a subset of the vertices of G

Independent-Set

Exists a set of k vertices s.t. no two are neighbors of each other?



Invalid Certificate

Instance: a graph G and a threshold number k

Certificate: a subset of the vertices of G

Independent-Set Instance and Certificate

Independent-Set Instance and Certificate

Instance: graph G and natural k

Independent-Set Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Independent-Set Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Assertion for valid certificate C of (G,k):
Forall e in edges of G:

Independent-Set Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Assertion for valid certificate C of (G,k):

Forall e in edges of G:

Not (And one vertex of e in C

the other vertex of e in C)

INDEPENDENT-SET Instance and Certificate

Instance: graph G and natural k

Certificate: subset of vertices of G

Assertion for valid certificate C of (G,k):

Forall e in edges of G:

Not (And one vertex of e in C

the other vertex of e in C)

and

Size of C >= k

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3)$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3)$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3)$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4) \checkmark$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4) \checkmark$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4) \checkmark$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4) \checkmark$$

Invalid Certificate

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{T} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3)$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{F} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4)$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{F} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4) \checkmark$$

$$(x_2 \vee \neg x_3 \vee \neg x_4)$$

Valid or Not?

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{F} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT – Mother of All NP-Problems

Exists true/false assignment of the variable satisfying all clauses?

$$(\neg x_1 \vee x_2 \vee x_3) \checkmark$$

$$(x_1 \vee \neg x_2 \vee x_4) \checkmark$$

$$(x_2 \vee \neg x_3 \vee \neg x_4) \checkmark$$

Valid Certificate

Instance: A Boolean formula in 3-conjunctive normal form (CNF)

$$x_1 \rightsquigarrow \mathrm{F} \qquad x_2 \rightsquigarrow \mathrm{F} \qquad x_3 \rightsquigarrow \mathrm{F} \qquad x_4 \rightsquigarrow \mathrm{F}$$

Certificate: Assignment from variables of the CNF to Boolean

3-SAT Instance and Certificate

Instance: 3CNF formula Phi

3-SAT Instance and Certificate

Instance: 3CNF formula Phi

Certificate: mapping from variables of Phi to Booleans

3-SAT Instance and Certificate

Instance: 3CNF formula Phi

Certificate: mapping from variables of Phi to Booleans

Assertion for valid certificate C of Phi:

Forall c in clauses of Phi:

Exists (literal l in c s.t

l is satisfied under c)