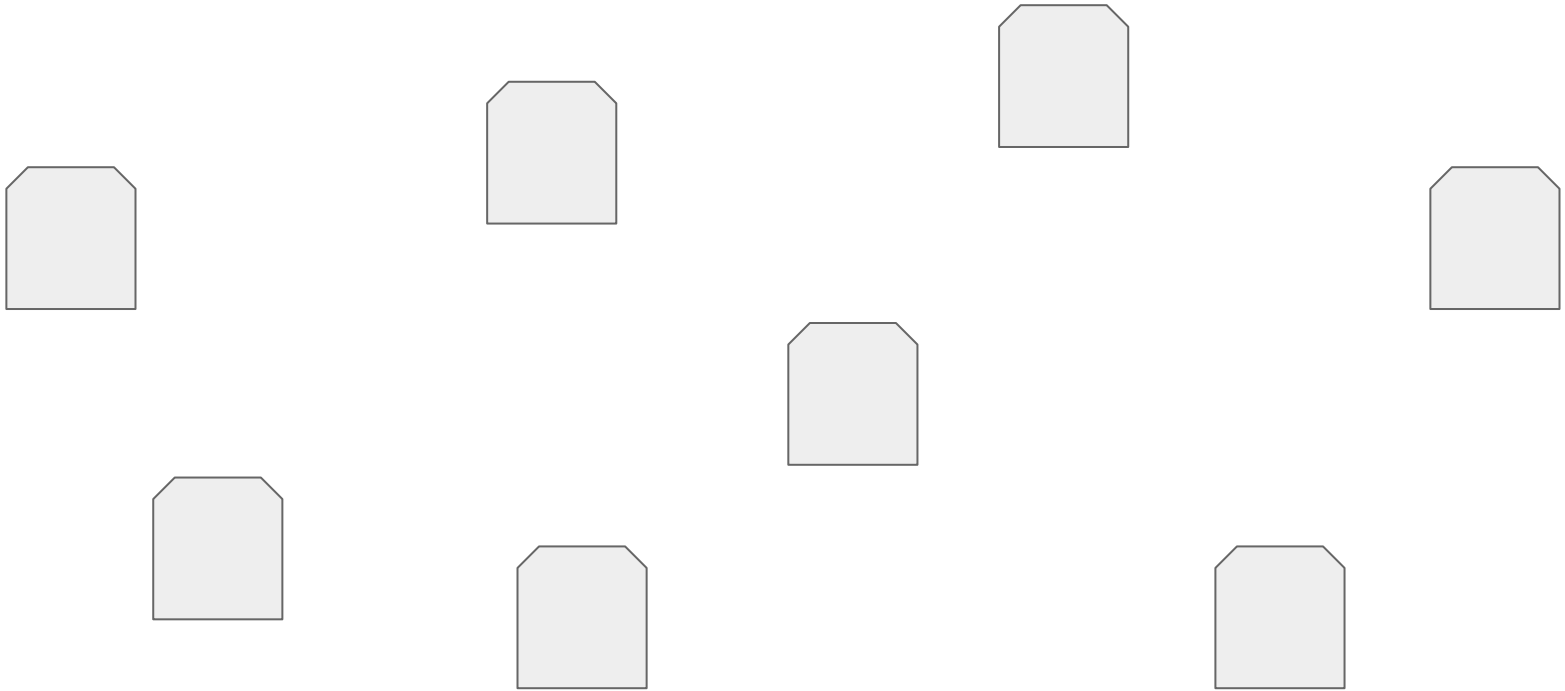
How hard is it to solve this?

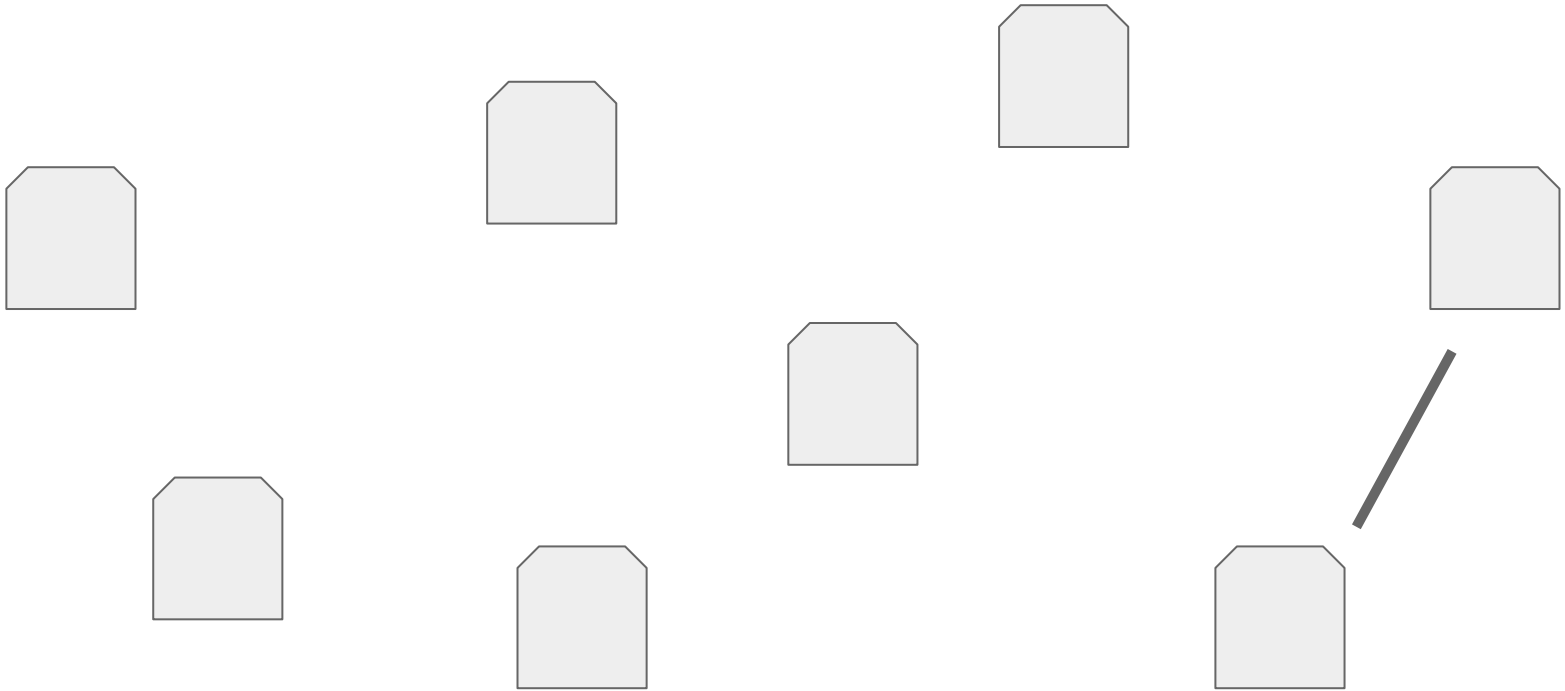
Prerona Chatterjee

NISER Bhubaneswar

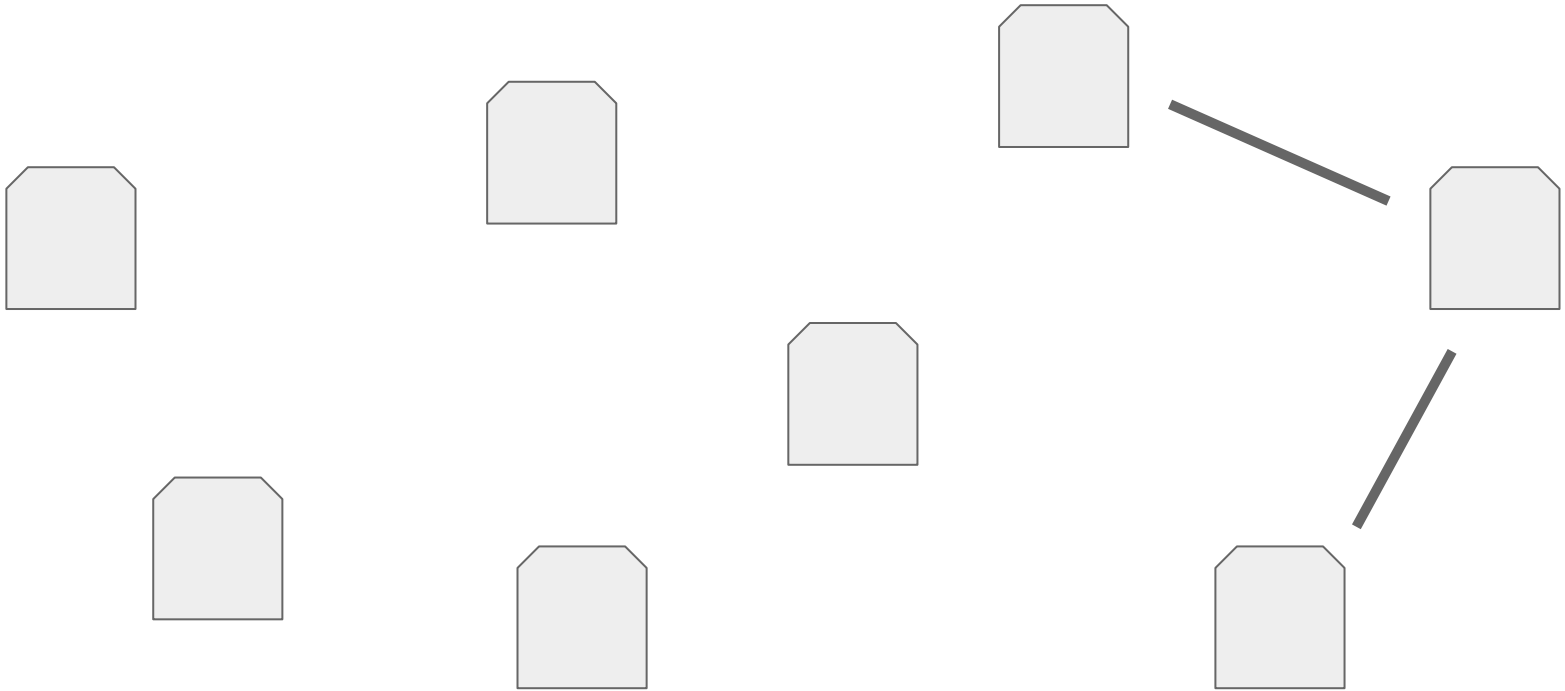
Connecting villages



Connecting villages



Connecting villages



Connecting villages

- Want to connect the villages.

Connecting villages

- Want to connect the villages.
- There should be a (possibly indirect) road between any two village.

Connecting villages

- Want to connect the villages.
- There should be a (possibly indirect) road between any two village.
- The cost of constructing a direct road between any two village is ₹10,000.

Connecting Villages

- Want to connect the villages.
- There should be a (possibly indirect) road between any two village.
- The cost of constructing a direct road between any two village is ₹10,000.

Can each pair of villages be connected with ₹50,000?

Connecting villages

Can each pair of villages be connected with ₹50,000?

Connecting villages

Can each pair of villages be connected with ₹50,000?

- If yes, how?
- If no, why not? How much budget would be enough?

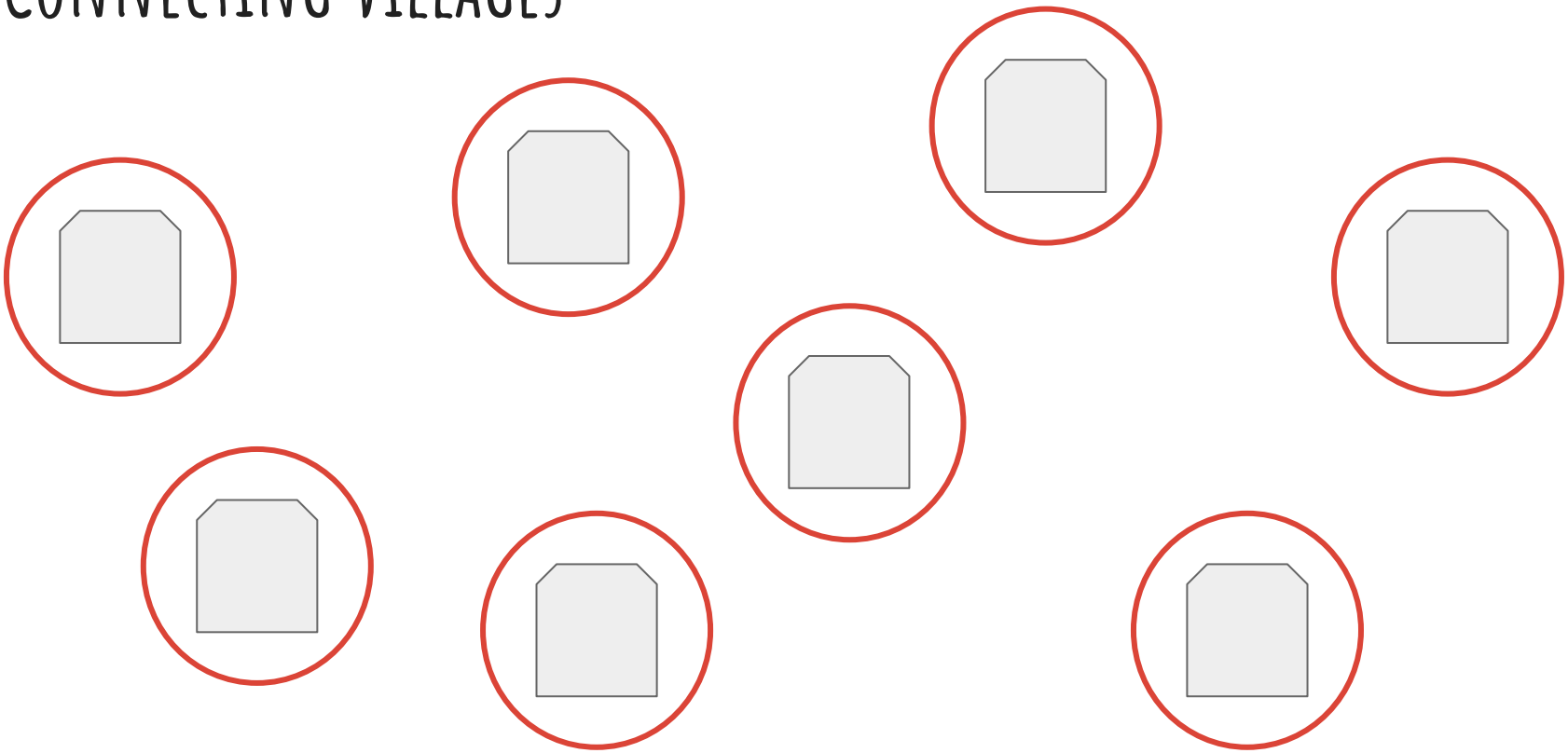
Connecting villages

Can each pair of villages be connected with ₹50,000?

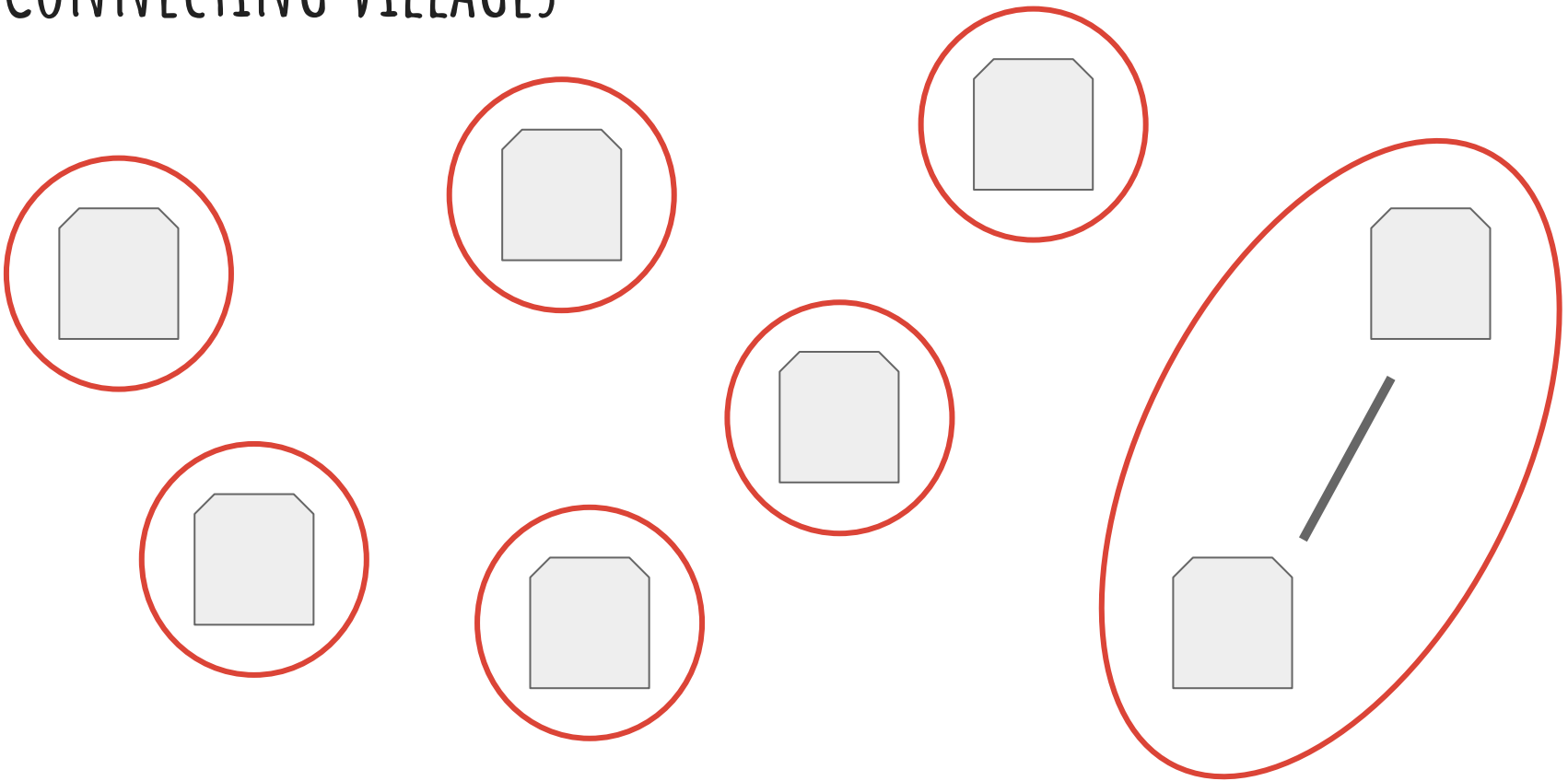
- If yes, how?
- If no, why not? How much budget would be enough?

₹70,000 is enough. But is it necessary?

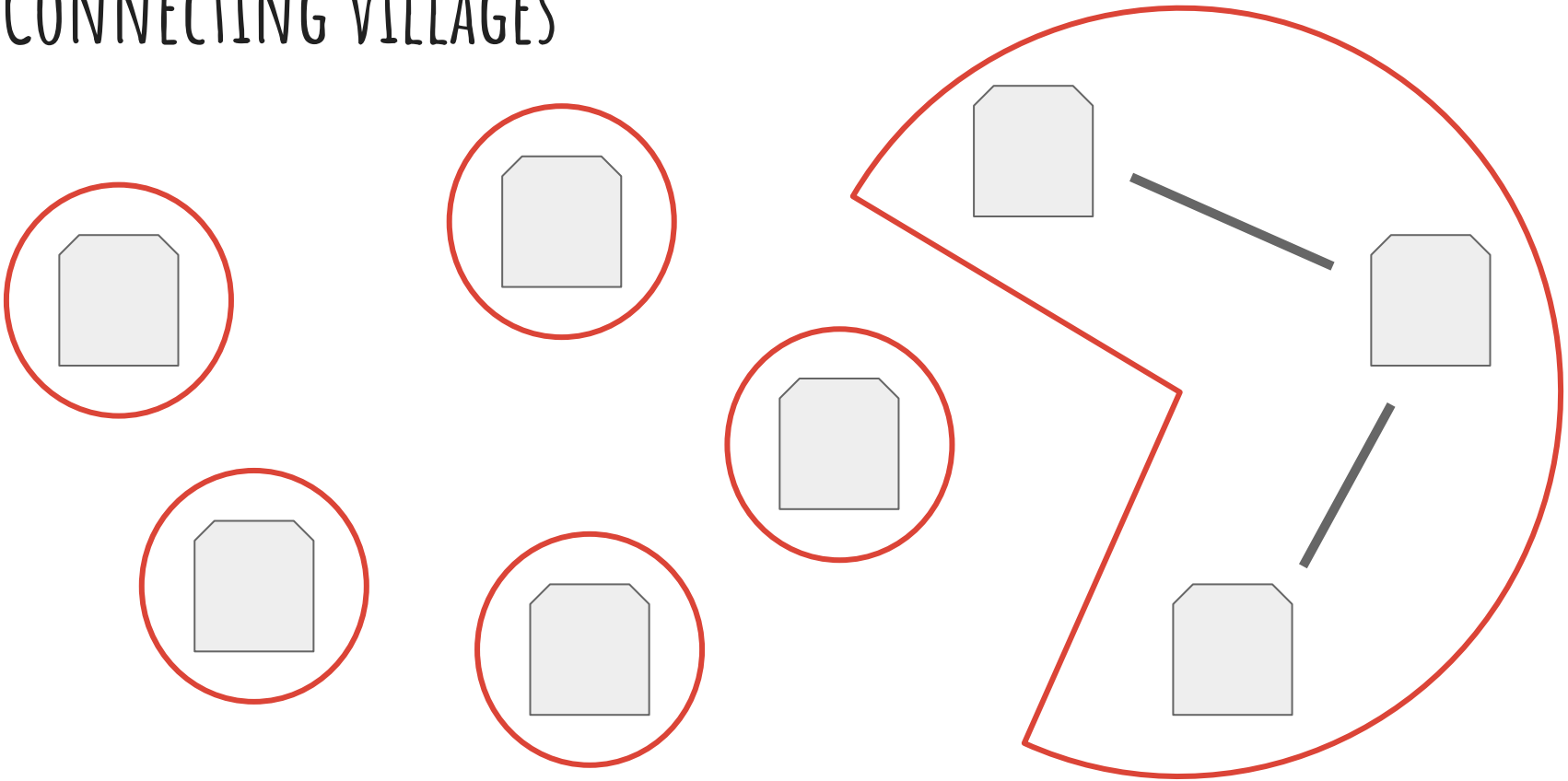
Connecting villages



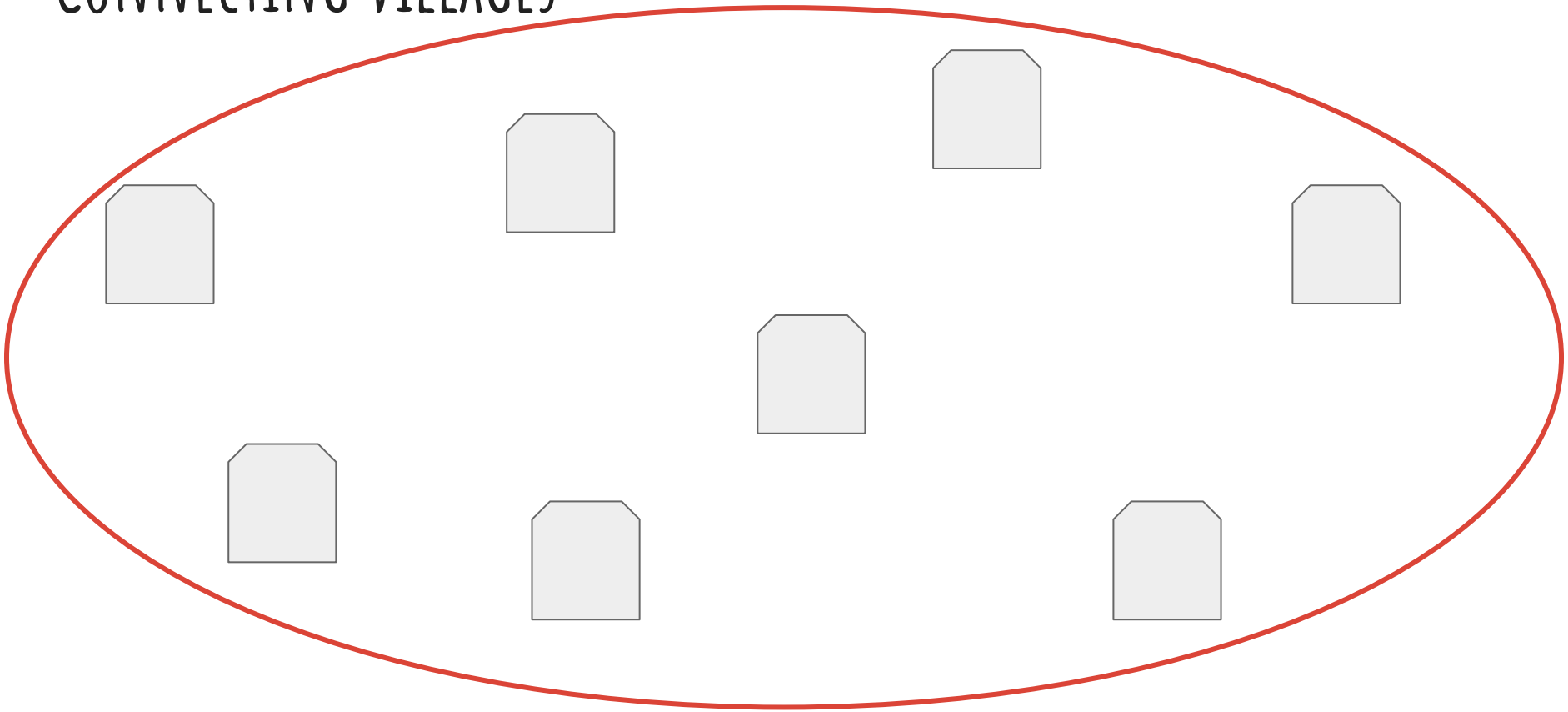
Connecting villages



Connecting villages



Connecting villages



Connecting villages

The number of components at the beginning = n

Connecting villages

The number of components at the beginning = n

The number of components at the end = 1

Connecting villages

The number of components at the beginning = n

The number of components at the end = 1

At each step the number of components reduces by at most 1

Connecting villages

The number of components at the beginning = n

The number of components at the end = 1

At each step the number of components reduces by at most 1

Therefore the number of steps is at least n-1

Connecting villages

The number of components at the beginning = n

The number of components at the end = 1

At each step the number of components reduces by at most 1

Therefore the number of steps is at least n-1

₹70,000 is indeed necessary.

How is proving that something is hard helpful?

Knowing the limitations

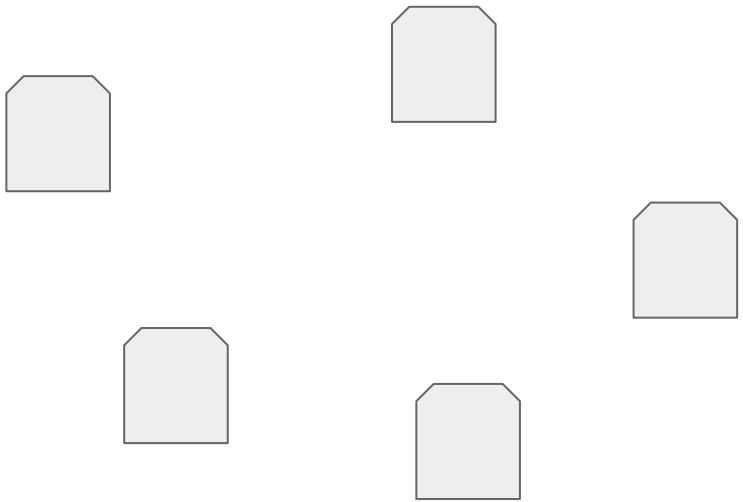
The problem in its full generality might be hard to solve. But…

Knowing the limitations

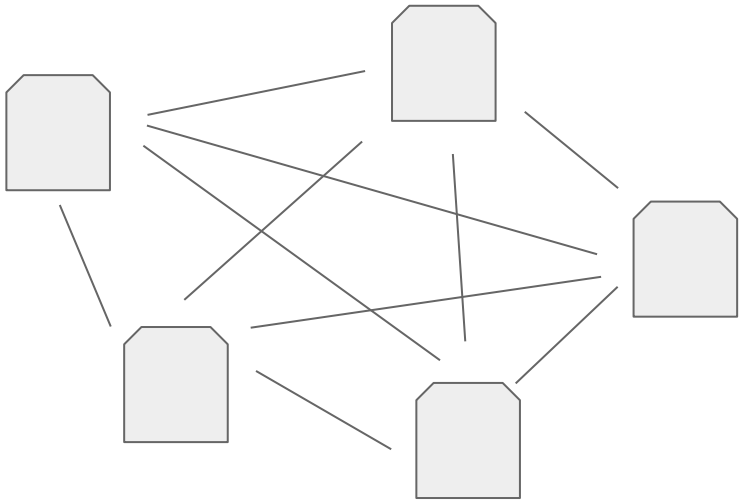
The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?

The travelling salesperson

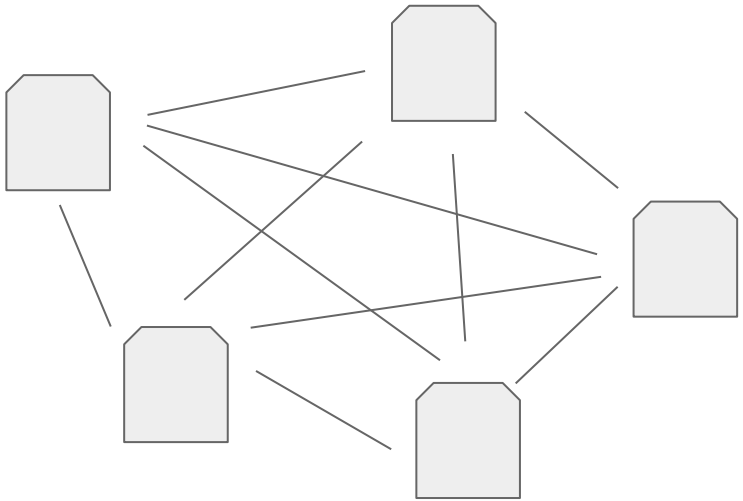


The travelling salesperson



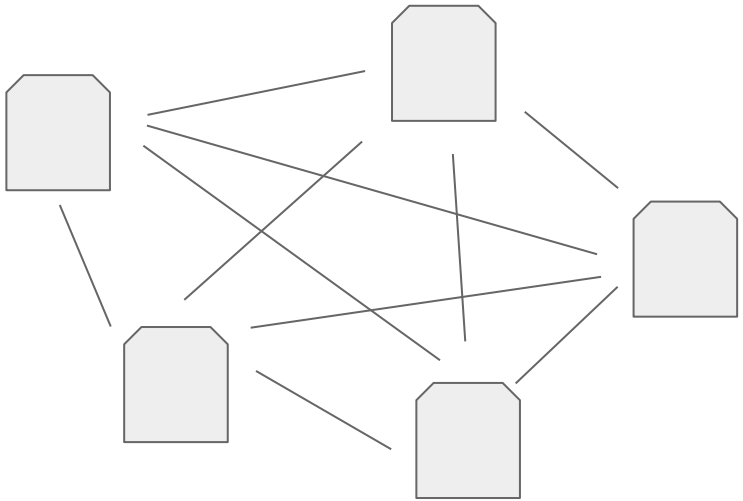
- All the villages are connected to each other by a direct road.

The travelling salesperson



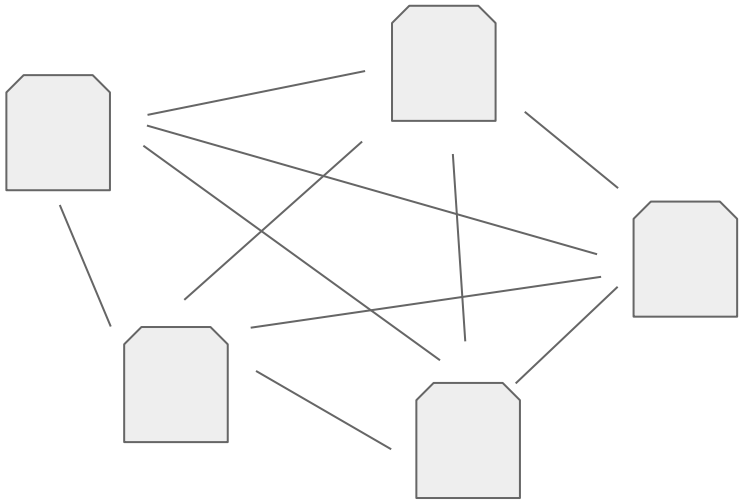
- All the villages are connected to each other by a direct road.
- Different (direct) roads are of different lengths.

The travelling salesperson



- All the villages are connected to each other by a direct road.
- Different (direct) roads are of different lengths.
- Salesperson wants to visit each village exactly once and return to the village they started from.

The travelling salesperson



- All the villages are connected to each other by a direct road.
- Different (direct) roads are of different lengths.
- Salesperson wants to visit each village exactly once and return to the village they started from.
- Find the route which minimises the distance.

Knowing the limitations

Approximation algorithms

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?

Approximation algorithms

- It is a way of dealing with optimisation problems that is believed to be hard.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?

Approximation algorithms

- It is a way of dealing with optimisation problems that is believed to be hard.
- Guarantees efficiency but does not guarantee the most optimal solution.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?

Approximation algorithms

- It is a way of dealing with optimisation problems that is believed to be hard.
- Guarantees efficiency but does not guarantee the most optimal solution.
- Guarantees to give a solution that is close to the optimal.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?

Approximation algorithms

- It is a way of dealing with optimisation problems that is believed to be hard.
- Guarantees efficiency but does not guarantee the most optimal solution.
- Guarantees to give a solution that is close to the optimal.

MTSP: Believed to be hard to solve exactly.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?

Approximation algorithms

- It is a way of dealing with optimisation problems that is believed to be hard.
- Guarantees efficiency but does not guarantee the most optimal solution.
- Guarantees to give a solution that is close to the optimal.

MTSP: Believed to be hard to solve exactly.

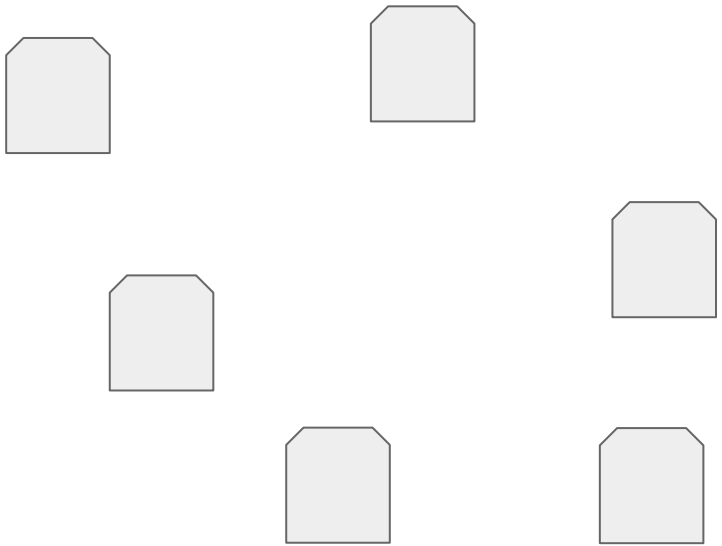
has an efficient 1.5-approximation algo.

Knowing the limitations

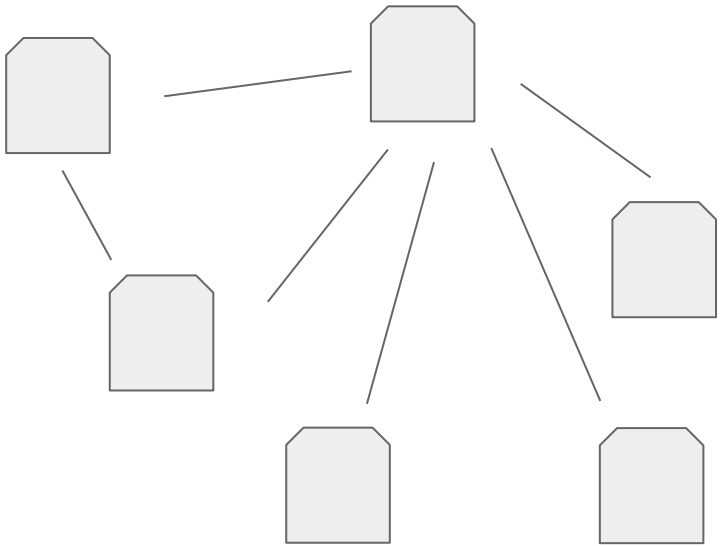
The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?

k-Vertex Cover

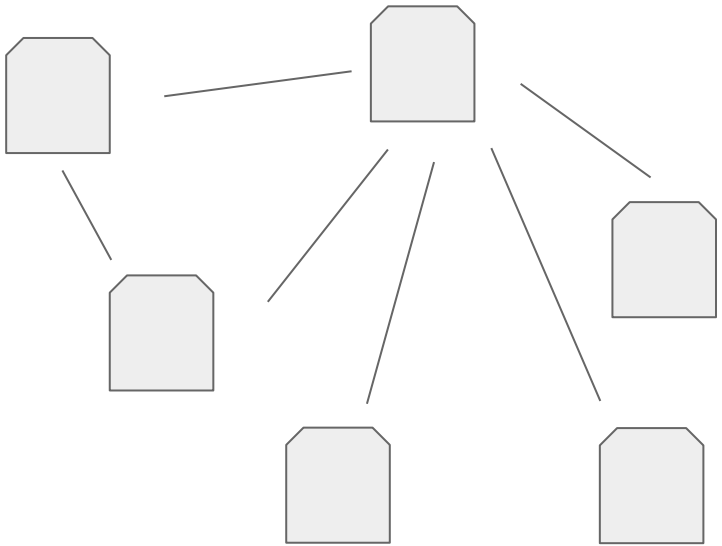


k-Vertex Cover



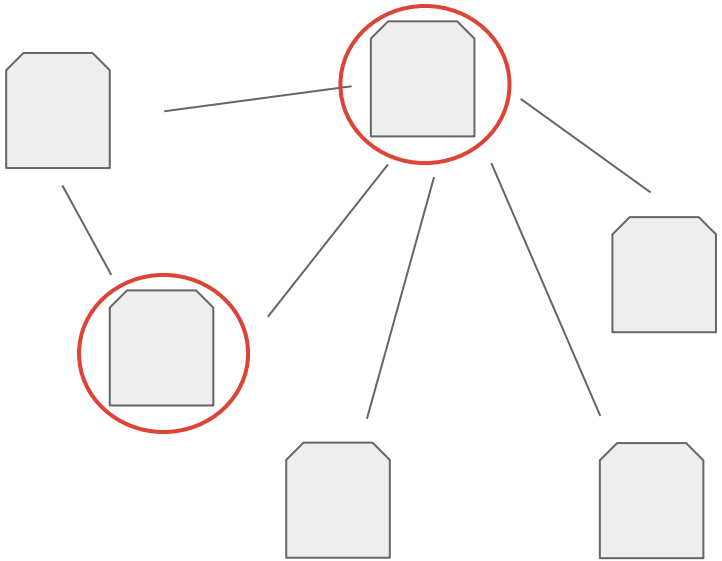
- All the villages are connected to each other, but not necessarily by a direct road.

k-Vertex Cover



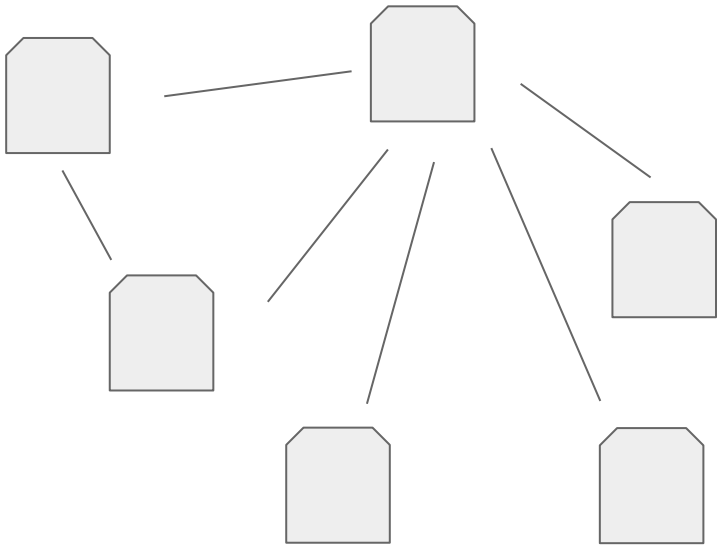
- All the villages are connected to each other, but not necessarily by a direct road.
- Want to place police stations in some villages so that each road is guarded by some police station.

k-Vertex Cover



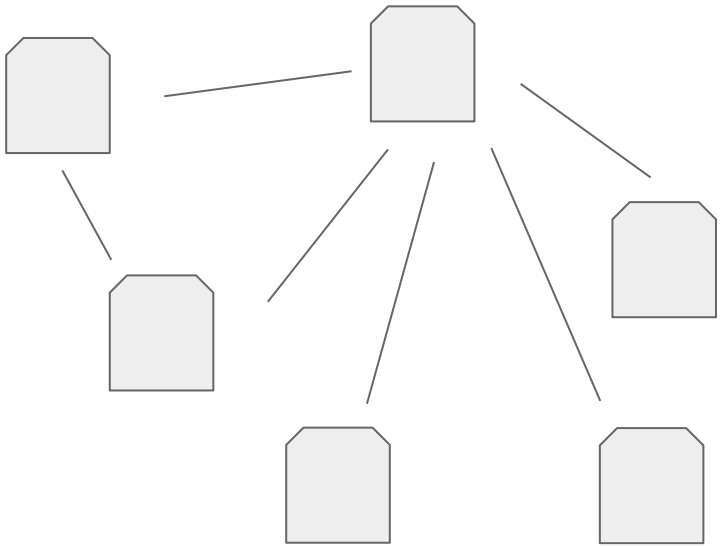
- All the villages are connected to each other, but not necessarily by a direct road.
- Want to place police stations in some villages so that each road is guarded by some police station.

k-Vertex Cover



- All the villages are connected to each other, but not necessarily by a direct road.
- Want to place police stations in some villages so that each road is guarded by some police station.
- For a given k, find out if it is possible to do this with only k police stations.

k-Vertex Cover



- All the villages are connected to each other, but not necessarily by a direct road.
- Want to place police stations in some villages so that each road is guarded by some police station.
- For a given k, find out if it is possible to do this with only k police stations. Output the villages they should be constructed in.

Knowing the limitations

Parameterised algorithms

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?

Parameterised algorithms

- It is a way of dealing with problems where one of the input parameters is expected to be small.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?

Parameterised algorithms

- It is a way of dealing with problems where one of the input parameters is expected to be small.
- Guarantees efficiency in cases when the parameter in question is indeed small.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?

Parameterised algorithms

- It is a way of dealing with problems where one of the input parameters is expected to be small.
- Guarantees efficiency in cases when the parameter in question is indeed small.

k-vertex cover: Believed to be hard if both k and n are large.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?

Parameterised algorithms

- It is a way of dealing with problems where one of the input parameters is expected to be small.
- Guarantees efficiency in cases when the parameter in question is indeed small.

k-vertex cover: Believed to be hard if both k and n are large.

has an efficient algo if k is small.

Secretly adding numbers

- Create groups of 5.
- Each person think of a two digit number.

Secretly adding numbers

- Create groups of 5.
- Each person think of a two digit number.

As a group, you want to find the sum of the numbers selected by your members.

Secretly adding numbers

- Create groups of 5.
- Each person think of a two digit number.

As a group, you want to find the sum of the numbers selected by your members.

As a group, you want to find the sum of the numbers selected by your members,

Secretly adding numbers

- Create groups of 5.
- Each person think of a two digit number.

As a group, you want to find the sum of the numbers selected by your members.

As a group, you want to find the sum of the numbers selected by your members, WITHOUT ANY OF YOU FINDING OUT THE NUMBERS SELECTED BY THE OTHERS.

Knowing the limitations

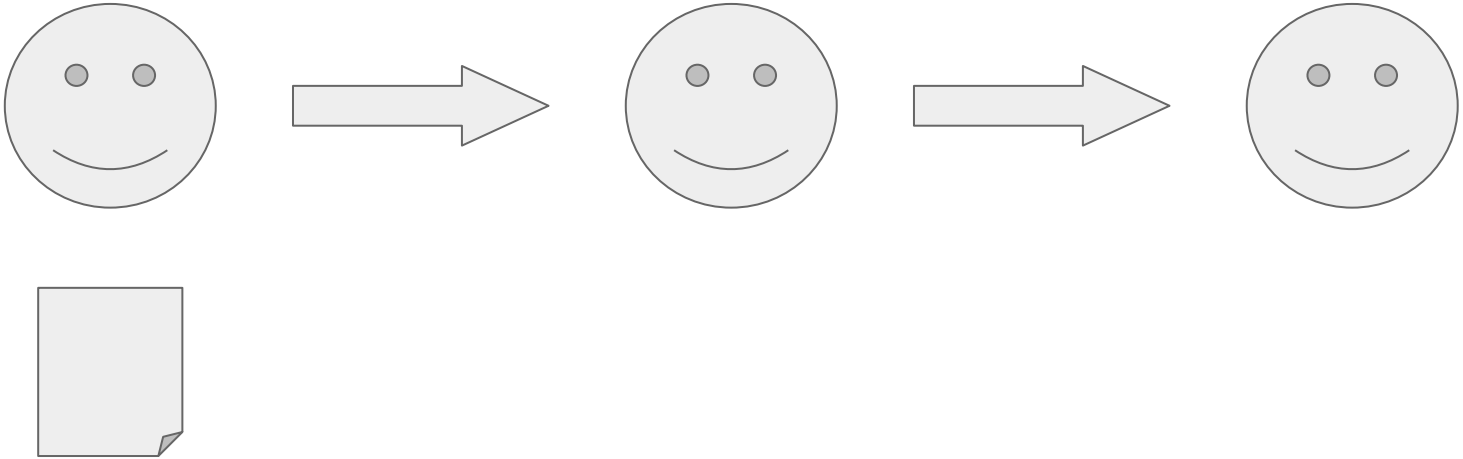
The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?
- Can the limitations in power of an adversary be used to come up with secure protocols?

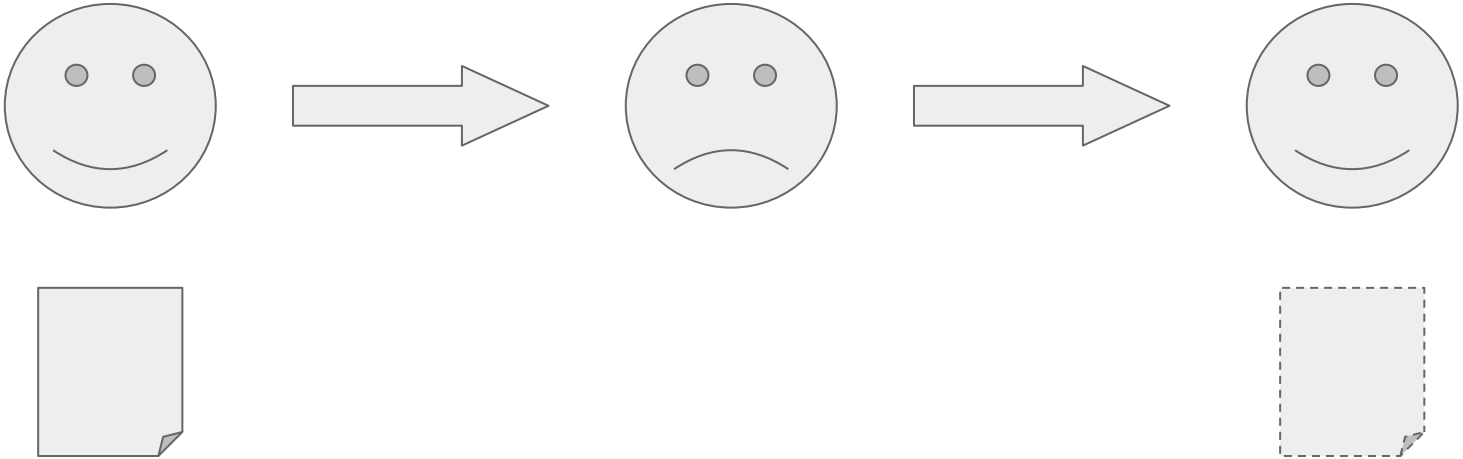
Sending secret messages

Sending secret messages

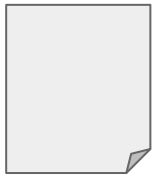
Sending secret messages

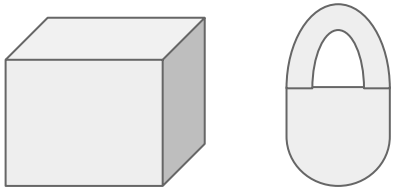


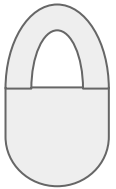
Sending secret messages



Sending secret messages







Sending secret messages

- A puts the note in the box and locks it. Gives it to B.

Sending secret messages

- A puts the note in the box and locks it. Gives it to B.
- B gives it to C.

Sending secret messages

- A puts the note in the box and locks it. Gives it to B.
- B gives it to C.
- C adds their lock as well. Gives it to B.

Sending secret messages

- A puts the note in the box and locks it. Gives it to B.
- B gives it to C.
- C adds their lock as well. Gives it to B.
- B gives it to A.

Sending secret messages

- A puts the note in the box and locks it. Gives it to B.
- B gives it to C.
- C adds their lock as well. Gives it to B.
- B gives it to A.
- A removes their lock using their key. Gives it to B.

Sending secret messages

- A puts the note in the box and locks it. Gives it to B.
- B gives it to C.
- C adds their lock as well. Gives it to B.
- B gives it to A.
- A removes their lock using their key. Gives it to B.
- B gives it to C.

Sending secret messages

- A puts the note in the box and locks it. Gives it to B.
- B gives it to C.
- C adds their lock as well. Gives it to B.
- B gives it to A.
- A removes their lock using their key. Gives it to B.
- B gives it to C.
- C unlocks their lock using their key to reveal note.

Sending secret messages

- A puts the note in the box and locks it. Gives it to B.
- B gives it to C.
- C adds their lock as well. Gives it to B.
- B gives it to A.
- A removes their lock using their key. Gives it to B.
- B gives it to C.
- C unlocks their lock using their key to reveal note.

This protocol is secure assuming B does not have any equipment to break the lock.

Knowing the limitations

Cryptography

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?
- Can the limitations in power of an adversary be used to come up with secure protocols?

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?
- Can the limitations in power of an adversary be used to come up with secure protocols?

Cryptography

Study of techniques for secure communication in the presence of adversarial behavior.

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?
- Can the limitations in power of an adversary be used to come up with secure protocols?

Cryptography

Study of techniques for secure communication in the presence of adversarial behavior.

Diffie-Helman Key Exchange

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?
- Can the limitations in power of an adversary be used to come up with secure protocols?

Cryptography

Study of techniques for secure communication in the presence of adversarial behavior.

Diffie-Helman Key Exchange

F : Easy to compute but hard to invert function

Knowing the limitations

The problem in its full generality might be hard to solve. But…

- Is it necessary to find the exact solution?
- Does the actual problem you want to solve have additional restrictions?
- Can the limitations in power of an adversary be used to come up with secure protocols?

Cryptography

Study of techniques for secure communication in the presence of adversarial behavior.

Diffie-Helman Key Exchange

F : Easy to compute but hard to invert function

G : Func. s.t. G(x,F(y)) = G(y,F(x))

Theoretical Computer Science

- A Problem that needs to be solved.

- A Problem that needs to be solved.
- A Mathematical Model that describes the abilities/restrictions of the solver.

- A Problem that needs to be solved.
- A Mathematical Model that describes the abilities/restrictions of the solver.

Study the amount of resources needed by the model to solve the problem.

I am more interested in solving the problem

I AM MORE INTERESTED IN SOLVING THE PROBLEM

- Finding factors of a given number.
- Finding the shortest path between two vertices in a graph.
- Finding the next best move in a chess game.
- Finding out if the given image is that of a dog or a cat.
- Predicting the next word in a sentence.
- .. ..

I am more interested in solving the problem

- Finding factors of a given number.
- Finding the shortest path between two vertices in a graph.
- Finding the next best move in a chess game.
- Finding out if the given image is that of a dog or a cat.
- Predicting the next word in a sentence.
- .. ..

I am more interested in understanding the model

I am more interested in solving the problem

- Finding factors of a given number.
- Finding the shortest path between two vertices in a graph.
- Finding the next best move in a chess game.
- Finding out if the given image is that of a dog or a cat.
- Predicting the next word in a sentence.
- .. ..

I am more interested in understanding the model

- Communication models.
  - two-party/multi-party
  - compromised channels/uncompromised channels
  - broadcast/point-to-point
- Computers (a.k.a Turing machines)
- Quantum Computers.
- Circuits.
- Proof Systems.
- Prover-Verifier Games.
- .. ..

Algebraic Complexity Theory

Understanding the complexity of algebraic problems

Understanding the complexity of algebraic problems

- Finding factors of a given number/polynomial.
- Computing the determinant of a matrix.
- Multiplying two numbers/polynomials/matrices.
- Finding out if a given set of vectors are linearly independent.
- Finding an annihilator for a given set of polynomials.
- .. ..

Understanding the complexity of algebraic problems

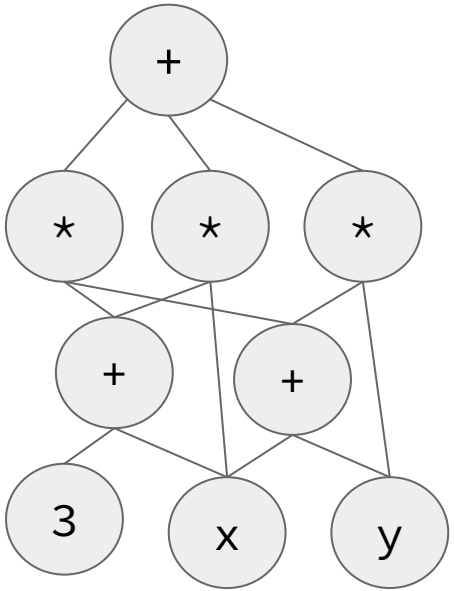
- Finding factors of a given number/polynomial.
- Computing the determinant of a matrix.
- Multiplying two numbers/polynomials/matrices.
- Finding out if a given set of vectors are linearly independent.
- Finding an annihilator for a given set of polynomials.
- .. ..

understanding the power of algebraic models

Understanding the complexity of algebraic problems

- Finding factors of a given number/polynomial.
- Computing the determinant of a matrix.
- Multiplying two numbers/polynomials/matrices.
- Finding out if a given set of vectors are linearly independent.
- Finding an annihilator for a given set of polynomials.
- .. ..

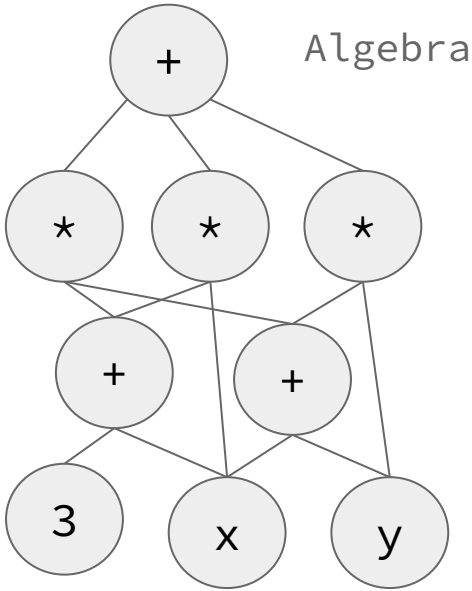
understanding the power of algebraic models



Understanding the complexity of algebraic problems

- Finding factors of a given number/polynomial.
- Computing the determinant of a matrix.
- Multiplying two numbers/polynomials/matrices.
- Finding out if a given set of vectors are linearly independent.
- Finding an annihilator for a given set of polynomials.
- .. ..

understanding the power of algebraic models

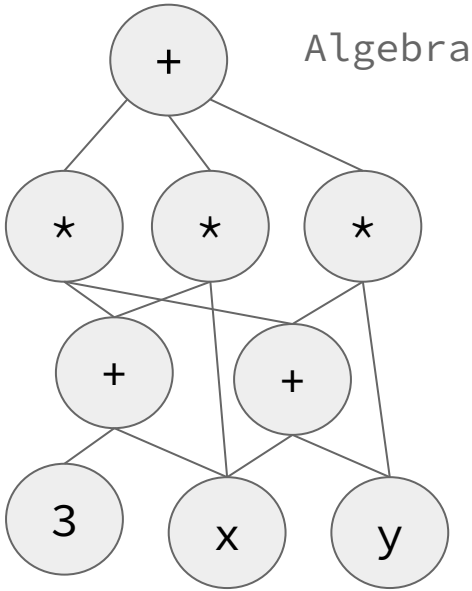


Algebraic Circuits

Understanding the complexity of algebraic problems

- Finding factors of a given number/polynomial.
- Computing the determinant of a matrix.
- Multiplying two numbers/polynomials/matrices.
- Finding out if a given set of vectors are linearly independent.
- Finding an annihilator for a given set of polynomials.
- .. ..

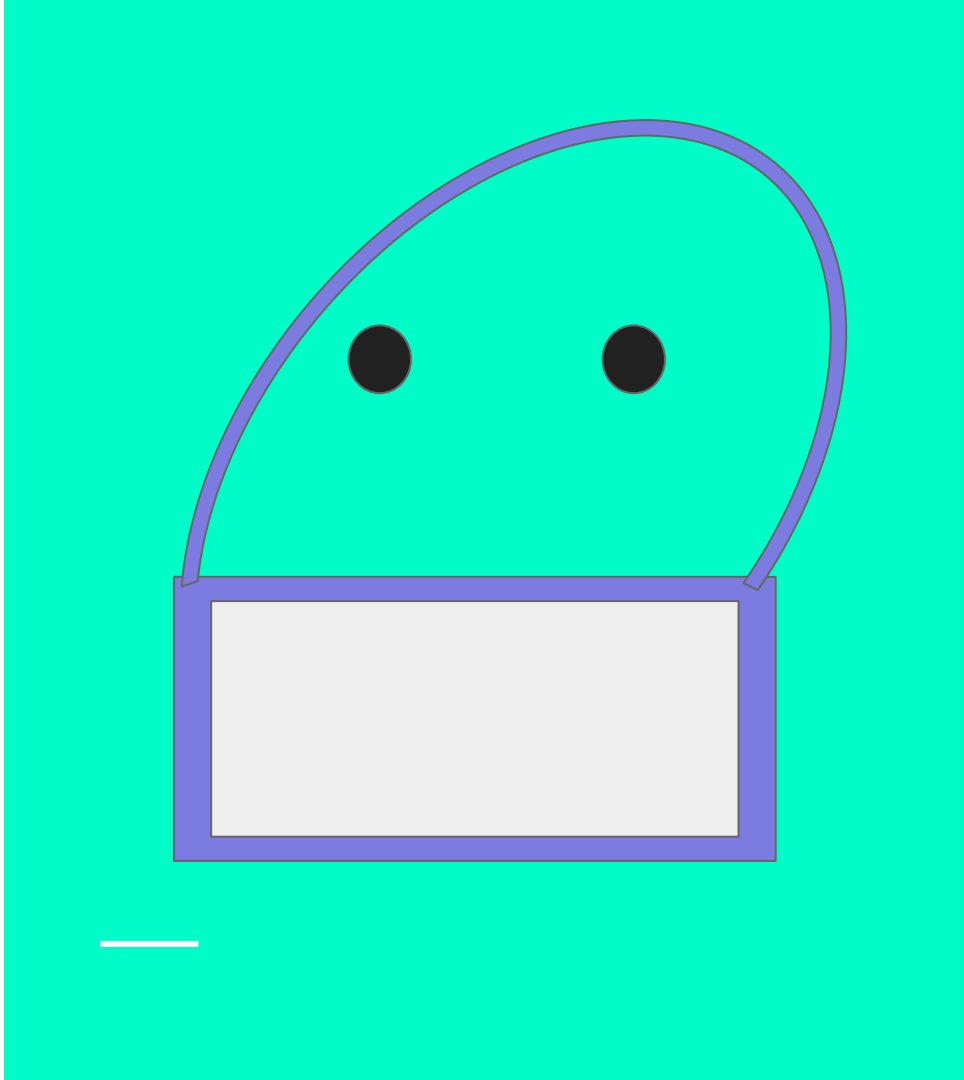
understanding the power of algebraic models



Algebraic Circuits

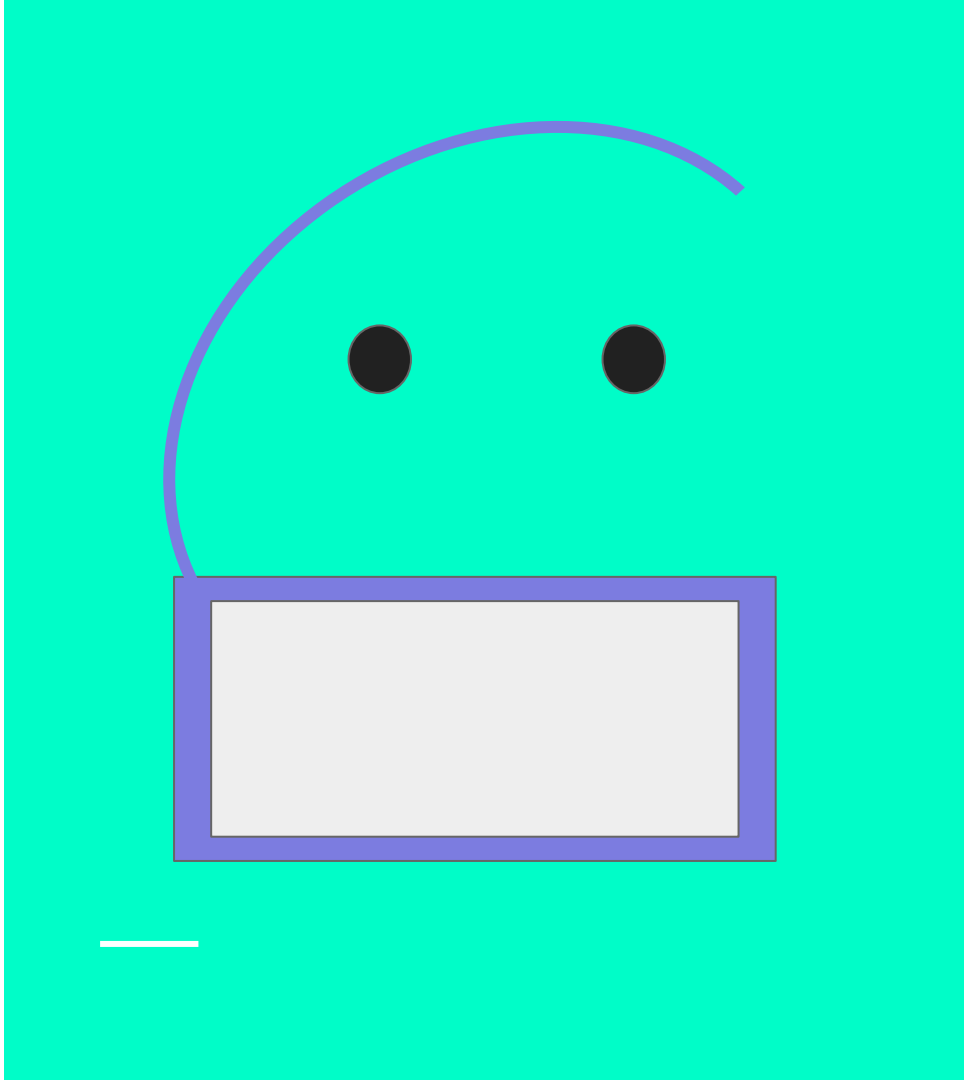
- Algebraic Formulas
- Algebraic Branching Programs
- BSS Model

The Picture Hanging Puzzle



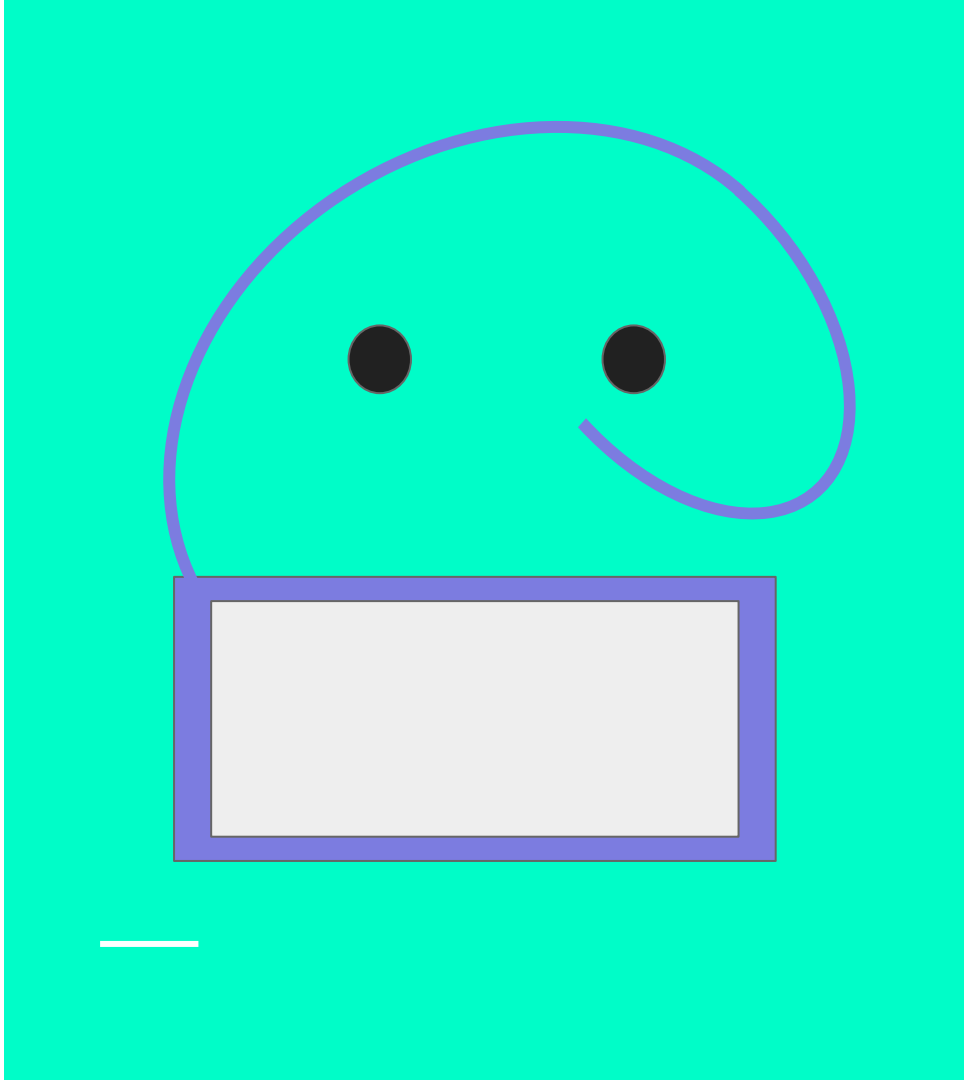
The Picture Hanging Puzzle

Solution: **A**



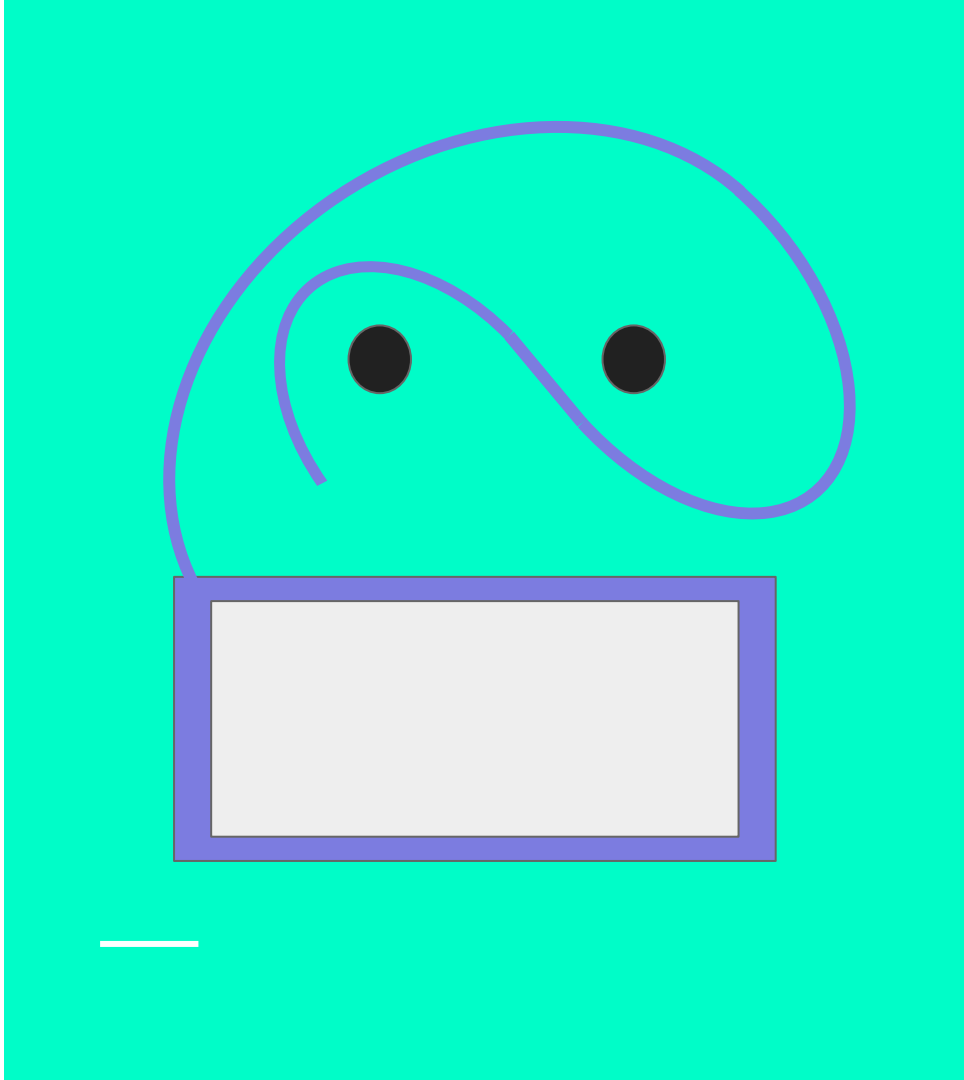
The Picture Hanging Puzzle

Solution: **AB**



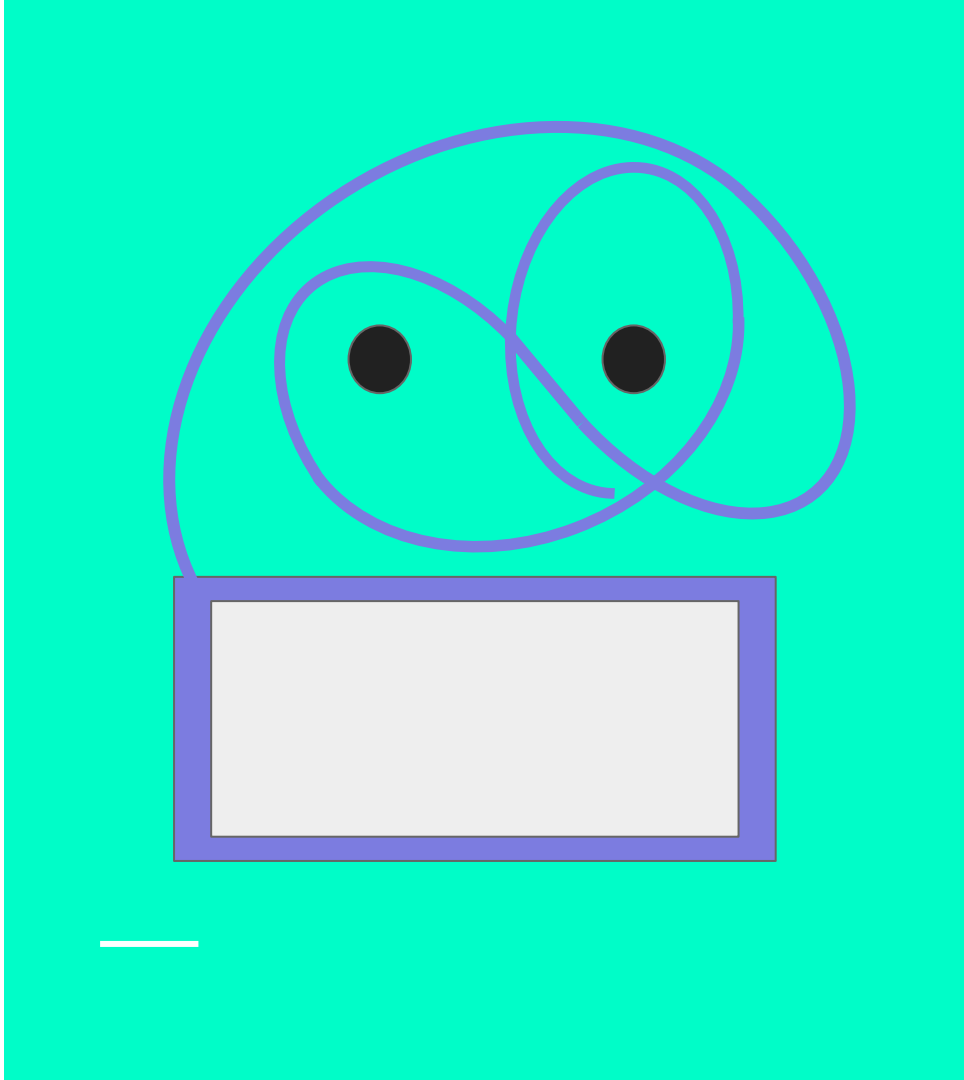
The Picture Hanging Puzzle

Solution: $\mathbf{ABA^{-1}}$



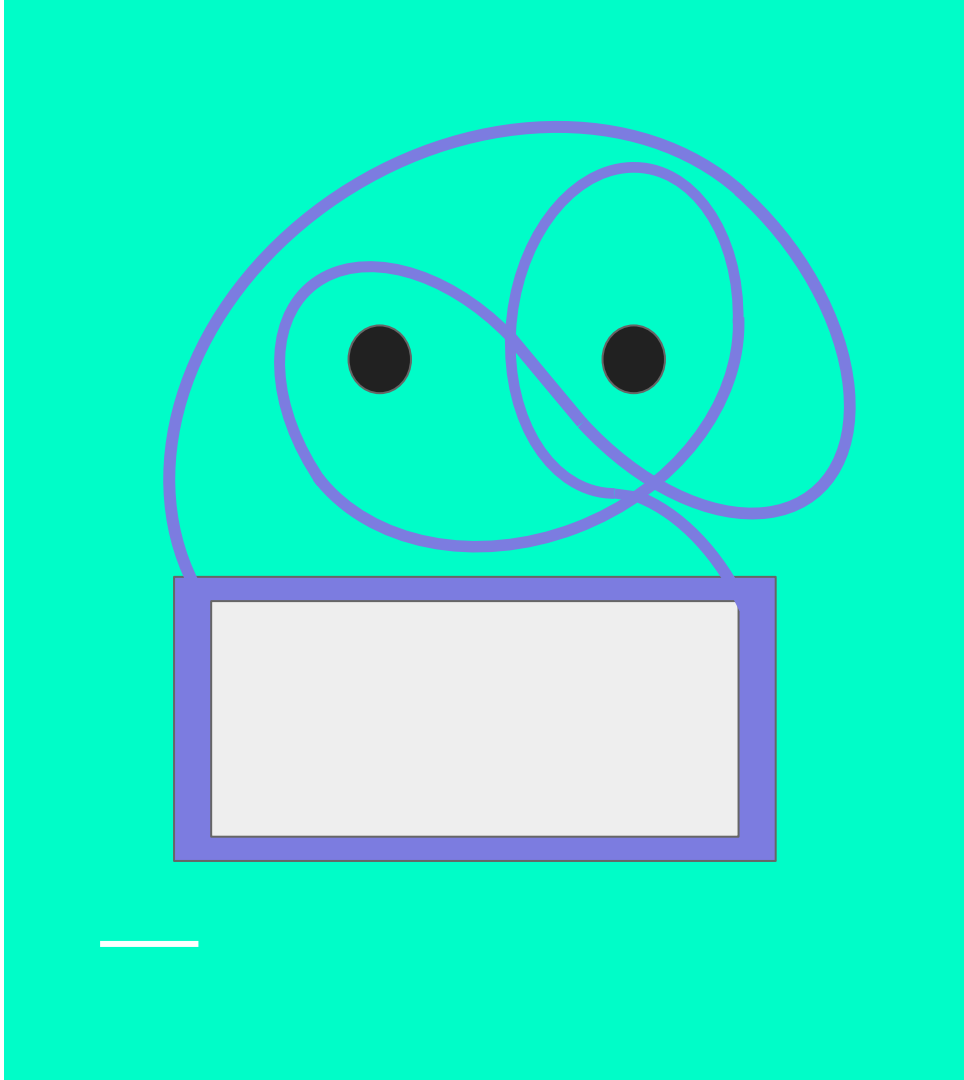
The Picture Hanging Puzzle

Solution: $\mathbf{ABA^{-1}B^{-1}}$



The Picture Hanging Puzzle

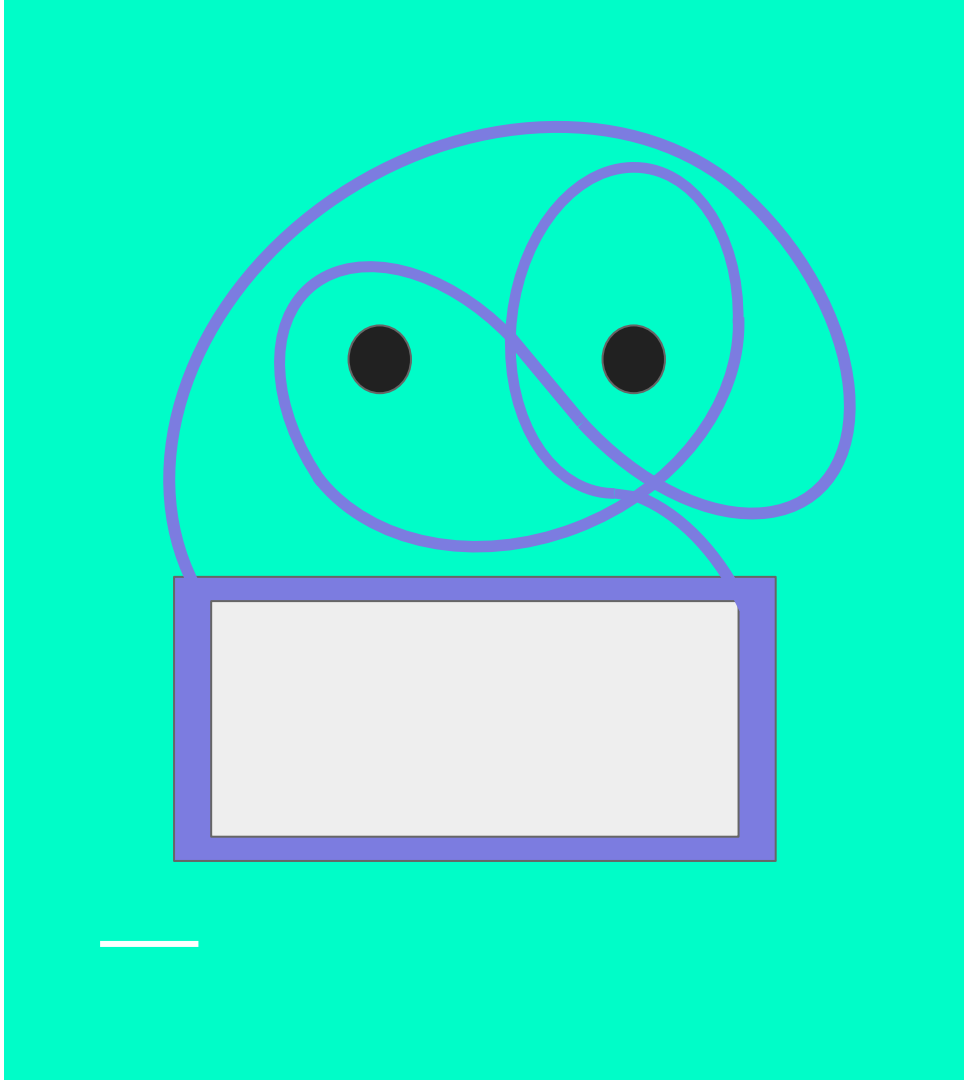
Solution: $ABA^{-1}B^{-1}$



The Picture Hanging Puzzle

Solution: $ABA^{-1}B^{-1}$

COMMUTATORS



Commutators

- Central concept in Galois Theory.

Commutators

- Central concept in Galois Theory.
- Solving a Rubik’s Cube.

Commutators

- Central concept in Galois Theory.
- Solving a Rubik’s Cube.
- Proving that Quintic equations have no closed form solutions.

Commutators

- Central concept in Galois Theory.
- Solving a Rubik’s Cube.
- Proving that Quintic equations have no closed form solutions.
- Proving that a regular heptagon can not be constructed using only a compass and a ruler.

Commutators

- Central concept in Galois Theory.
- Solving a Rubik’s Cube.
- Proving that Quintic equations have no closed form solutions.
- Proving that a regular heptagon can not be constructed using only a compass and a ruler.
- Proving the algebraic formulas are equivalent to width-3 algebraic branching programs.

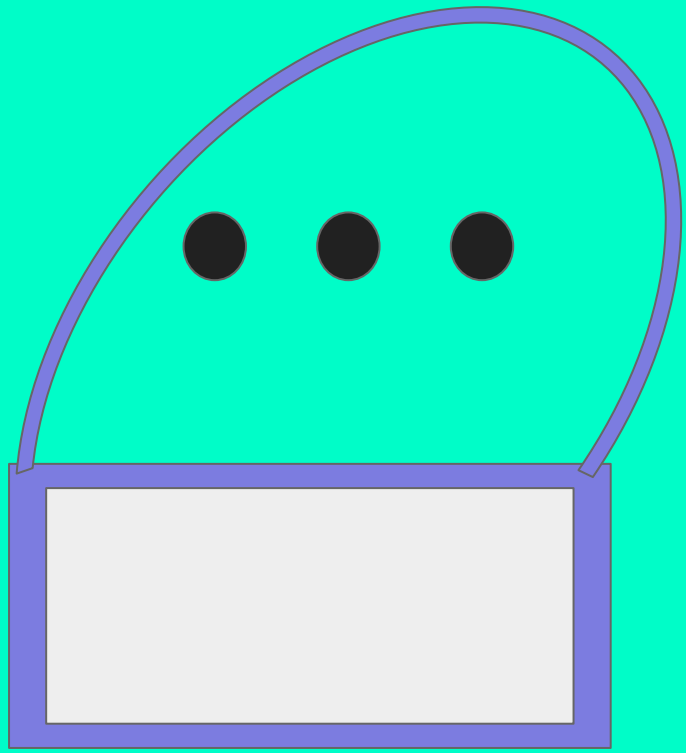
Commutators

- Central concept in Galois Theory.
- Solving a Rubik’s Cube.
- Proving that Quintic equations have no closed form solutions.
- Proving that a regular heptagon can not be constructed using only a compass and a ruler.
- Proving the algebraic formulas are equivalent to width-3 algebraic branching programs.

Check out: Chai and Why?(Dec 20, 2020) by Ramprasad Saptharishi

“Commutators! Hanging pictures and solving Rubik’s Cube”

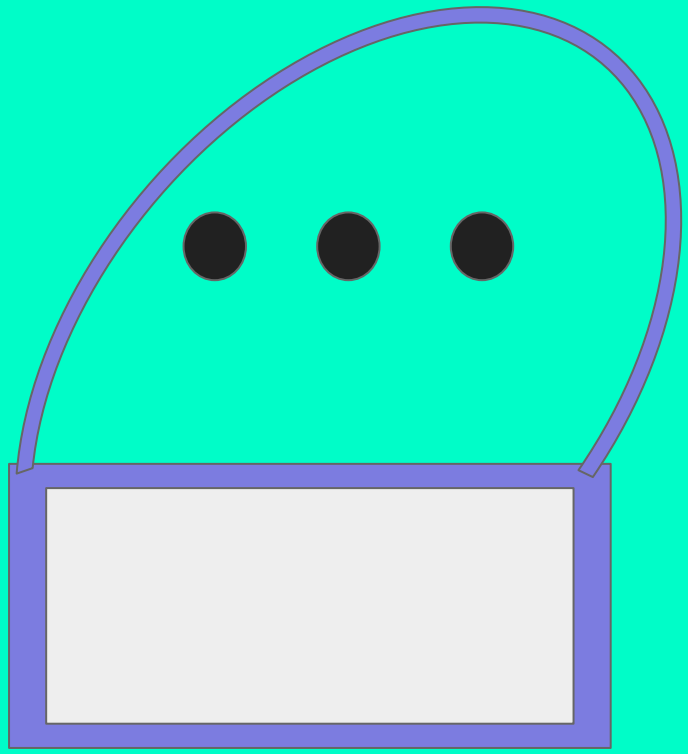
The Picture Hanging Puzzle



The Picture Hanging Puzzle

Solution

$$A(BCB^{-1}C^{-1})A^{-1}(CBC^{-1}B^{-1})$$



SCoS @ NISER

SCoS @ NISER

Secure Multiparty Computation

Secure multiparty computation (MPC) is a cryptographic protocol that **allows multiple parties to jointly compute a function** of their inputs while revealing as little as possible about those inputs.

Some of the applications of MPC are **online auctions, voting** etc.

Machine Learning

Machine Learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate in **predicting outcomes without being explicitly programmed to do so**.

Application of Machine learning includes **healthcare, natural language processing, recommendation systems.**

Data Clustering

Data clustering is the process of **grouping data points together based on their similarities**. Clustering is an unsupervised learning technique.

Application of data clustering includes **market segmentation, image segmentation, anomaly detection** and many more.

Algorithm Design

Algorithm Design refers to developing efficient algorithms to **solve computational problems and analysing their complexity**.

The aim is to devise algorithms that optimize time and space complexities, addressing challenges in various domains like **optimization, cryptography, artificial intelligence**, and more.

Complexity Theory

Complexity Theory is the study of different **computational models**.

Research in this area focuses on understanding the **power and limitations** of objects that model **real-world machines** with varied restrictions. Knowing the limitations of a model helps us devise **secure protocols** against them.

THANK YOU !

prerona.ch @ gmail.com

https://preronac.bitbucket.io/