Great Ideas in (theoretical) Computer Science

Sandeep Sen

Shiv Nadar Institution of Eminence

Nov 25, 2023

1. Coin tossing with social distance
2. How powerful are Computers
3. A measure of difficulty
4. Doing things randomly
5. Secret Communication
6. Exploiting Parallelism
7. TCS research in India

When does it qualify as Great

When does it qualify as Great

When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong.

- R. Buckminster Fuller

When does it qualify as Great

When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong.

- R. Buckminster Fuller

The mathematician's patterns, like the painter's or the poet's must be beautiful; the ideas like the colours or the words, must fit together in a harmonious way. Beauty is the first test: there is no permanent place in the world for ugly mathematics.

- GH Hardy, A Mathematician's Apology

When does it qualify as Great

When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong.

- R. Buckminster Fuller

The mathematician's patterns, like the painter's or the poet's must be beautiful; the ideas like the colours or the words, must fit together in a harmonious way. Beauty is the first test: there is no permanent place in the world for ugly mathematics.

- GH Hardy, A Mathematician's Apology

Long-lasting

When does it qualify as Great

When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong.

- R. Buckminster Fuller

The mathematician's patterns, like the painter's or the poet's must be beautiful; the ideas like the colours or the words, must fit together in a harmonious way. Beauty is the first test: there is no permanent place in the world for ugly mathematics.

- GH Hardy, A Mathematician's Apology

Long-lasting Ubiquity

When does it qualify as Great

When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong.

- R. Buckminster Fuller

The mathematician's patterns, like the painter's or the poet's must be beautiful; the ideas like the colours or the words, must fit together in a harmonious way. Beauty is the first test: there is no permanent place in the world for ugly mathematics.

- GH Hardy, A Mathematician's Apology

Long-lasting Ubiquity Simplicity

When does it qualify as Great

When I am working on a problem, I never think about beauty but when I have finished, if the solution is not beautiful, I know it is wrong.

- R. Buckminster Fuller

The mathematician's patterns, like the painter's or the poet's must be beautiful; the ideas like the colours or the words, must fit together in a harmonious way. Beauty is the first test: there is no permanent place in the world for ugly mathematics.

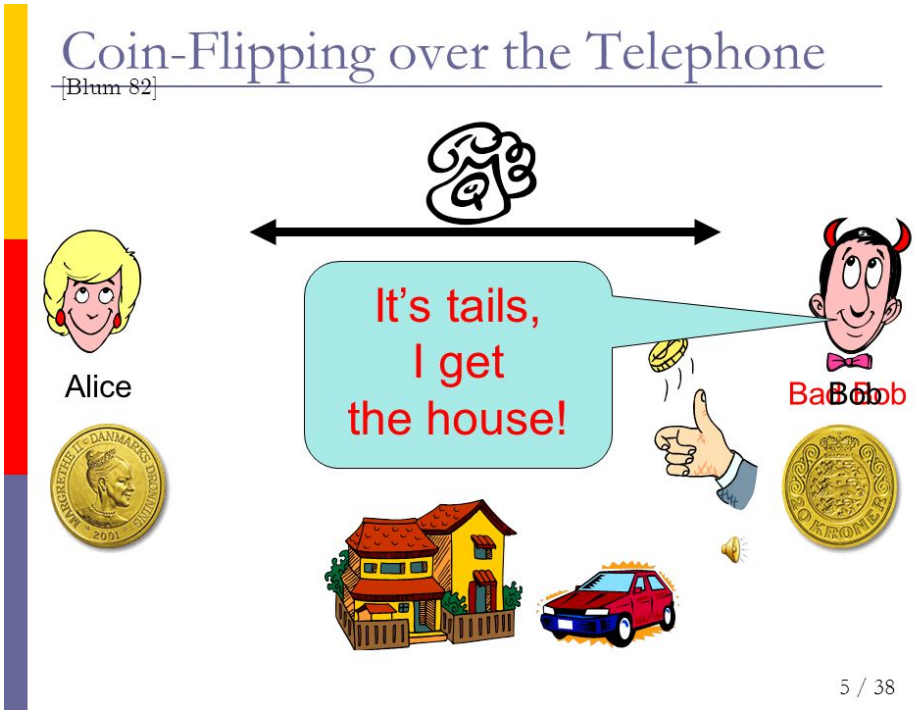
- GH Hardy, A Mathematician's Apology

Long-lasting Ubiquity Simplicity Beauty

Fairness using coin tossing

Fairness using coin tossing

Coin tossing over telephone



Alice tosses, Bob calls

Alice tosses, Bob calls

Nothing simultaneous when you are socially distant

Alice tosses, Bob calls

Nothing simultaneous when you are socially distant

Who goes first ?

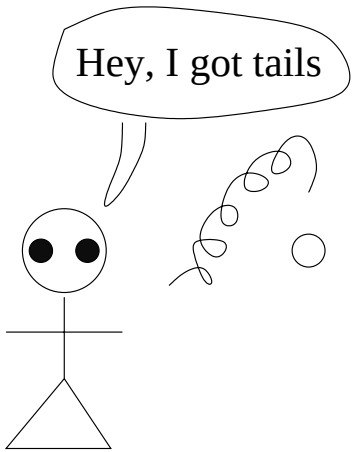
- If Alice tells Bob the outcome, Bob can cheat.

Alice tosses, Bob calls

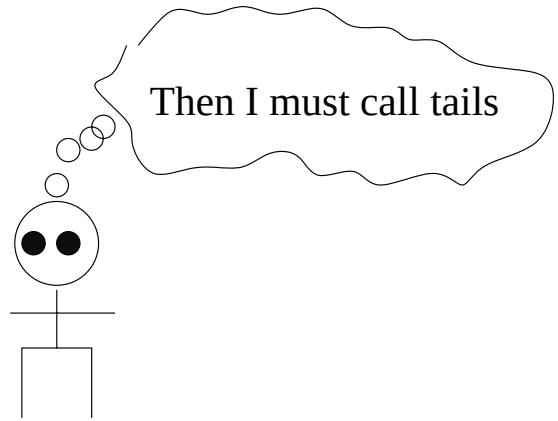
Nothing simultaneous when you are socially distant

Who goes first ?

- If Alice tells Bob the outcome, Bob can cheat.



ALICE



BOB

Alice tosses, Bob calls

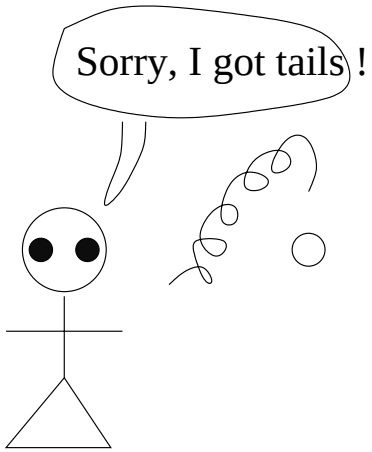
Who goes first ?

- If Bob calls first, then Alice can cheat.

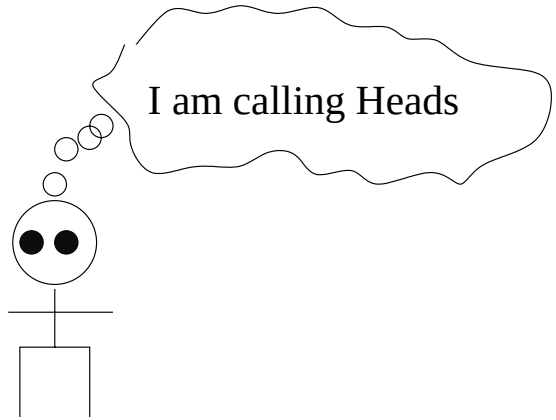
Alice tosses, Bob calls

Who goes first ?

- If Bob calls first, then Alice can cheat.



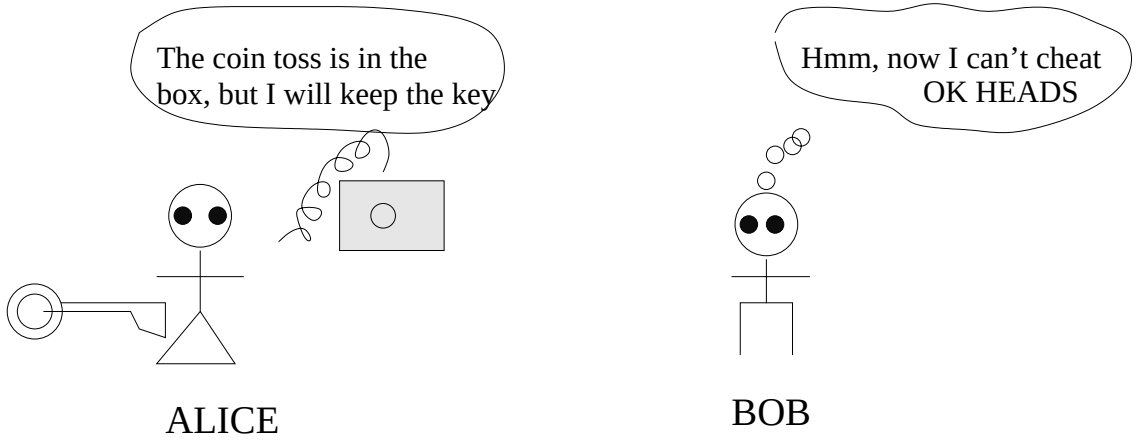
ALICE



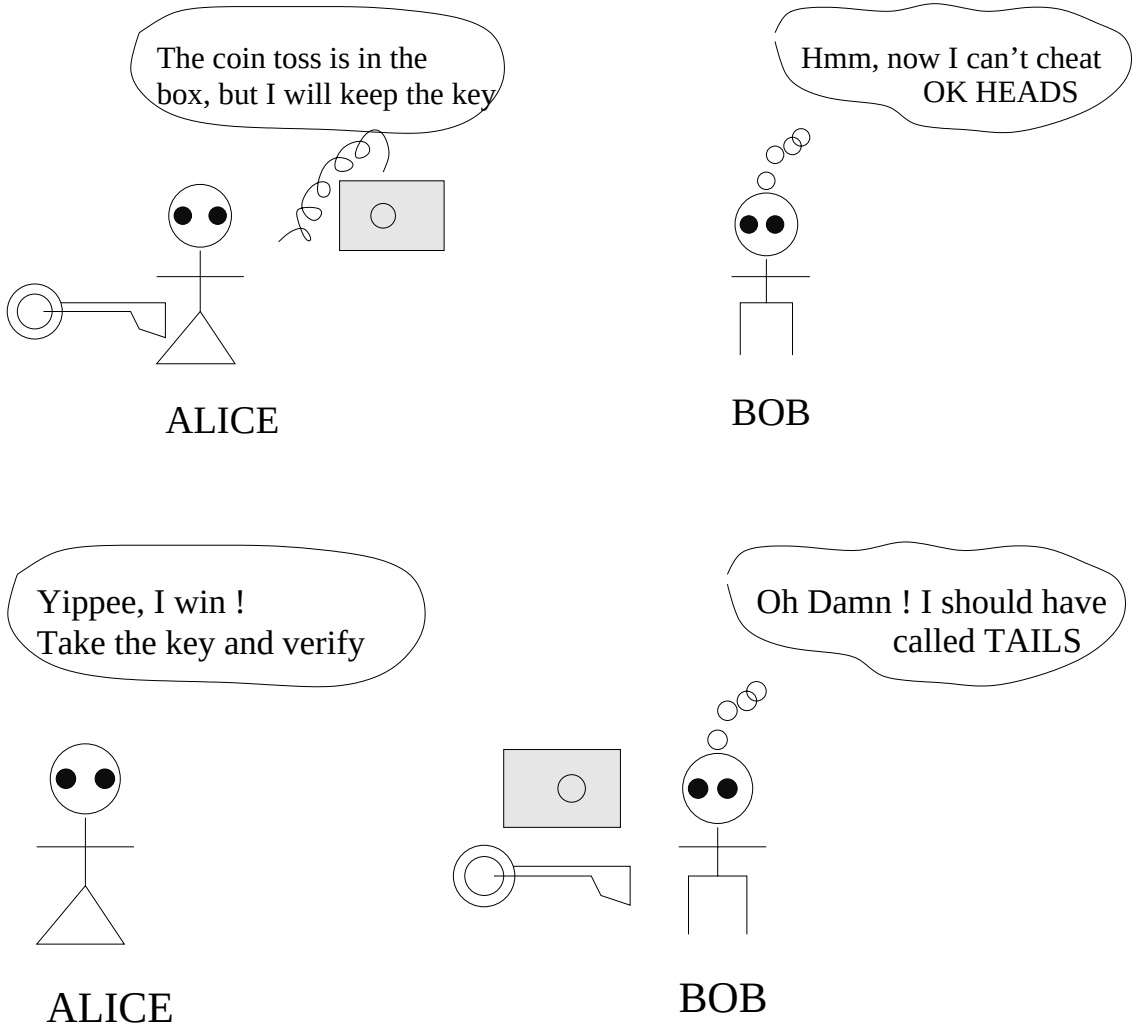
BOB

No one can cheat

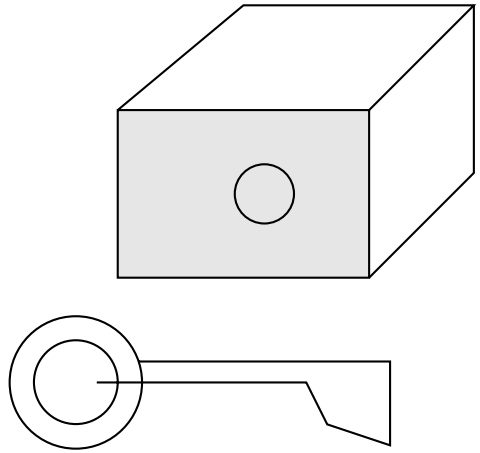
No one can cheat



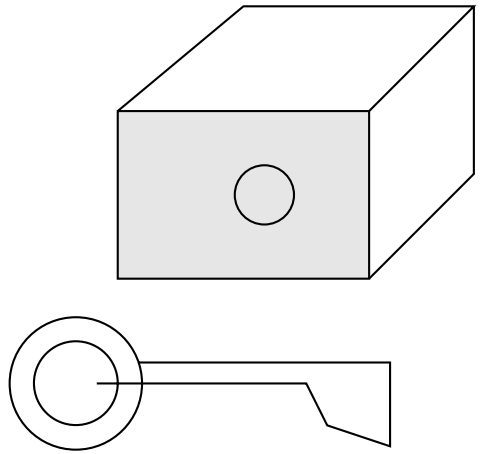
No one can cheat



Virtual lock and key



Virtual lock and key



Theory of One-way functions to realise such a functionality.

Why should one build a computer

Why should one build a computer

Building hardware itself is a miraculous technological feat - still progressing

Why should one build a computer

Building hardware itself is a miraculous technological feat - still progressing

Sophisticated software platforms based on Compilers and programming paradigms

Why should one build a computer

Building hardware itself is a miraculous technological feat - still progressing

Sophisticated software platforms based on Compilers and programming paradigms

Would we have invested so much in building computers if we didn't feel convinced about its potential and universality !!

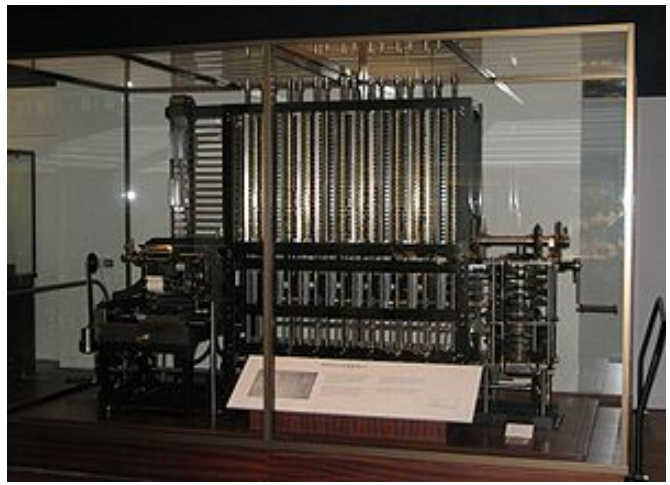
Some older machines

Leibnitz calculator 17th century

Some older machines

Leibnitz calculator 17th century

Babbage 19th century



Can there be a fundamentally superior Computer

Can there be a fundamentally superior Computer

Every year you have newer models of cars coming in - faster, slicker, more loaded.

Can there be a fundamentally superior Computer

Every year you have newer models of cars coming in - faster, slicker, more loaded.

So are laptops, cellphones etc.

Can there be a fundamentally superior Computer

Every year you have newer models of cars coming in - faster, slicker, more loaded.

So are laptops, cellphones etc.

What can Computers do

Can there be a fundamentally superior Computer

Every year you have newer models of cars coming in - faster, slicker, more loaded.

So are laptops, cellphones etc.

What can Computers do (or not do) from an intellectual standpoint ??

Speed is not pertinent for this question

Can Computers be used to solve anything ?

S No.	A	B
1	0	100
2	01	00
3	110	11

Can Computers be used to solve anything ?

S No.	A	B
1	0	100
2	01	00
3	110	11

3 , , ,

A sequence : 110

B sequence : 11

Can Computers be used to solve anything ?

S No.	A	B
1	0	100
2	01	00
3	110	11

3 , 2, ,

A sequence : 11001

B sequence : 1100

Can Computers be used to solve anything ?

S No.	A	B
1	0	100
2	01	00
3	110	11

3 , 2, 1 ,

A sequence : 110010
B sequence : 1100100

Can Computers be used to solve anything ?

S No.	A	B
1	0	100
2	01	00
3	110	11

3 , 2, 1 , 2

A sequence : 11001001

B sequence : 110010000

Can Computers be used to solve anything ?

S No.	A	B
1	0	100
2	01	00
3	110	11

3 , 2, 1 , 2

A sequence : 11001001
B sequence : 110010000

Post Correspondence Problem [PCP]

Is there a sequence (of any finite length) over $\{1, 2, 3\}$ such that if we concatenate the two sets of strings A, B, they become identical ?

Can Computers be taught to do all our Maths ?

It can add, multiply, find square roots, solve equations

Can Computers be taught to do all our Maths ?

It can add, multiply, find square roots, solve equations

Can it prove and discover Theorems ?

David Hilbert around 1900 conjectured so, and also posed a set of 23 outstanding problems in Mathematics.

Can Computers be taught to do all our Maths ?

It can add, multiply, find square roots, solve equations

Can it prove and discover Theorems ?

David Hilbert around 1900 conjectured so, and also posed a set of 23 outstanding problems in Mathematics.

But he was proved incorrect by Yuri Matiyasevich (10th Problem) !!
Does a given set of (diophantine) equations have integer solutions ?

Can Computers be taught to do all our Maths ?

It can add, multiply, find square roots, solve equations

Can it prove and discover Theorems ?

David Hilbert around 1900 conjectured so, and also posed a set of 23 outstanding problems in Mathematics.

But he was proved incorrect by Yuri Matiyasevich (10th Problem) !!
Does a given set of (diophantine) equations have integer solutions ?

Does this contradict Wiles/Fermat result ? $x^i + y^i = z^i \quad i > 2$

Limitations of logic

Computers can be thought of as executing a sequence of steps with some underlying logic

Limitations of logic

Computers can be thought of as executing a sequence of steps with some underlying logic

AS DEVISED BY THE PROGRAMMER

Limitations of logic

Computers can be thought of as executing a sequence of steps with some underlying logic

AS DEVISED BY THE PROGRAMMER

also referred to as an Algorithm

Limitations of logic

Computers can be thought of as executing a sequence of steps with some underlying logic
AS DEVISED BY THE PROGRAMMER
also referred to as an Algorithm

Goedel 1927

Logic is Incomplete - not powerful enough to discover all Theorems

Computers are universal

Church Turing thesis

The present paradigm of computing is the most powerful to compute whatever is possible to compute

Computers are universal

Church Turing thesis

The present paradigm of computing is the most powerful to compute whatever is possible to compute

The above thesis is INDEPENDENT of technology - no matter whether it is silicon based, or biological, molecular or quantum !

Computers are universal

Church Turing thesis

The present paradigm of computing is the most powerful to compute whatever is possible to compute

The above thesis is INDEPENDENT of technology - no matter whether it is silicon based, or biological, molecular or quantum !

So it made sense to go ahead and build one and not wait for a superior logical framework or technological advances.

Computers are universal

Church Turing thesis

The present paradigm of computing is the most powerful to compute whatever is possible to compute

The above thesis is INDEPENDENT of technology - no matter whether it is silicon based, or biological, molecular or quantum !

So it made sense to go ahead and build one and not wait for a superior logical framework or technological advances.

According to the thesis man and machine are not that different !

What is hard to solve ?

What is hard to solve ?

Partitioning numbers

$7, 25, 14, 9, 5, 18, 8$ Is there a *balanced* partitioning ?

What is hard to solve ?

Partitioning numbers

$7, 25, 14, 9, 5, 18, 8$ Is there a *balanced* partitioning ?

$14 + 9 + 5 + 8 + 7 = 25 + 18 = 43$

How about $20, 32, 15, 83, 61, 43, 9, 27, 55, 77, 35, 19, 52$

What is hard to solve ?

Partitioning numbers

$7, 25, 14, 9, 5, 18, 8$ Is there a *balanced* partitioning ?

$14 + 9 + 5 + 8 + 7 = 25 + 18 = 43$

How about $20, 32, 15, 83, 61, 43, 9, 27, 55, 77, 35, 19, 52$

If there is no such *partition* how do we figure out ?

What is hard to solve ?

Partitioning numbers

$7, 25, 14, 9, 5, 18, 8$ Is there a *balanced* partitioning ?

$14 + 9 + 5 + 8 + 7 = 25 + 18 = 43$

How about $20, 32, 15, 83, 61, 43, 9, 27, 55, 77, 35, 19, 52$

If there is no such *partition* how do we figure out ?

Games

Most of the interesting games are known to be compuationally *hard*.

What is hard to solve ?

Partitioning numbers

$7, 25, 14, 9, 5, 18, 8$ Is there a *balanced* partitioning ?

$14 + 9 + 5 + 8 + 7 = 25 + 18 = 43$

How about $20, 32, 15, 83, 61, 43, 9, 27, 55, 77, 35, 19, 52$

If there is no such *partition* how do we figure out ?

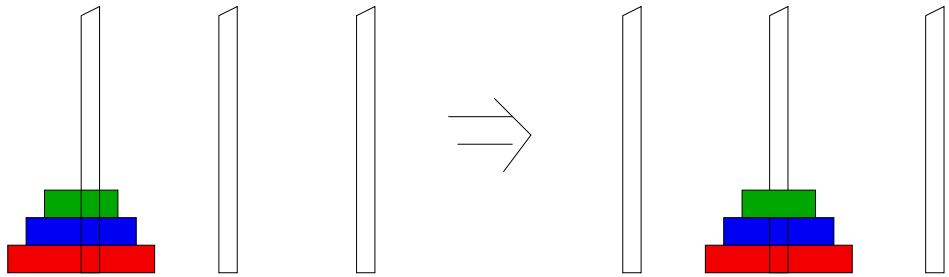
Games

Most of the interesting games are known to be compuationally *hard*.

Measure of difficulty

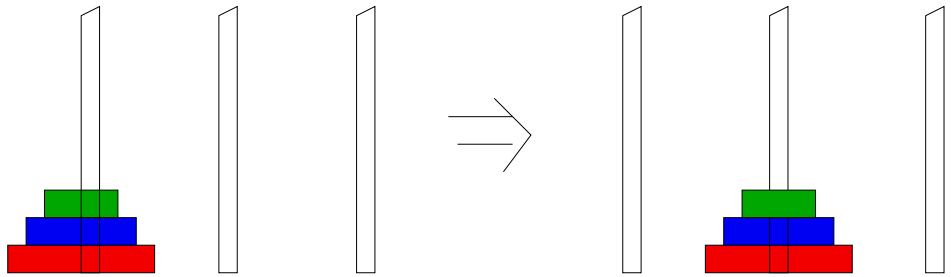
if the best algororithm to solve a problem takes exponential steps, then it is unrealistic for solving large sized problems - a hopeless situation.

Tower of Hanoi



▸ Click for video

Tower of Hanoi

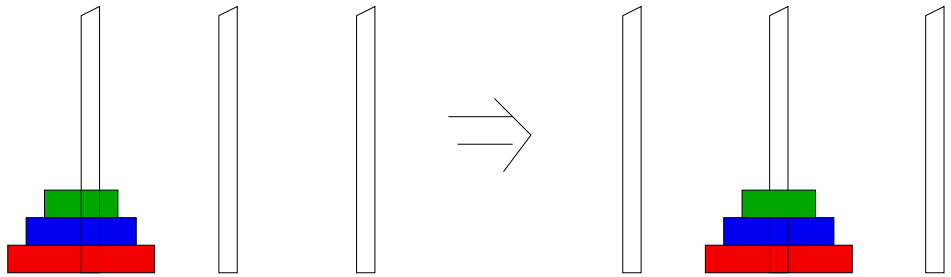


▸ Click for video

Time to solve 64 disks

If the priests were able to move disks at a rate of one per second, using the smallest number of moves it would take them $2^{64} - 1$ seconds or roughly 585 billion years to finish, which is about 42 times the current age of the Universe.

Tower of Hanoi



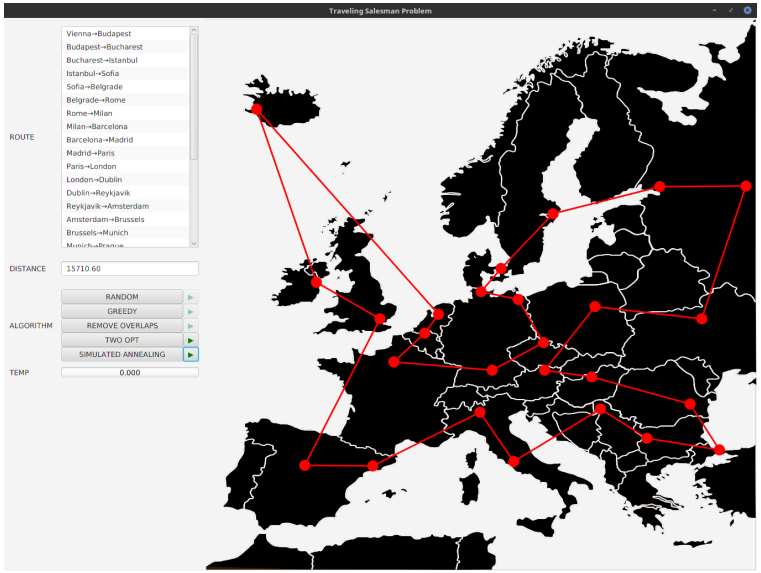
▸ Click for video

Time to solve 64 disks

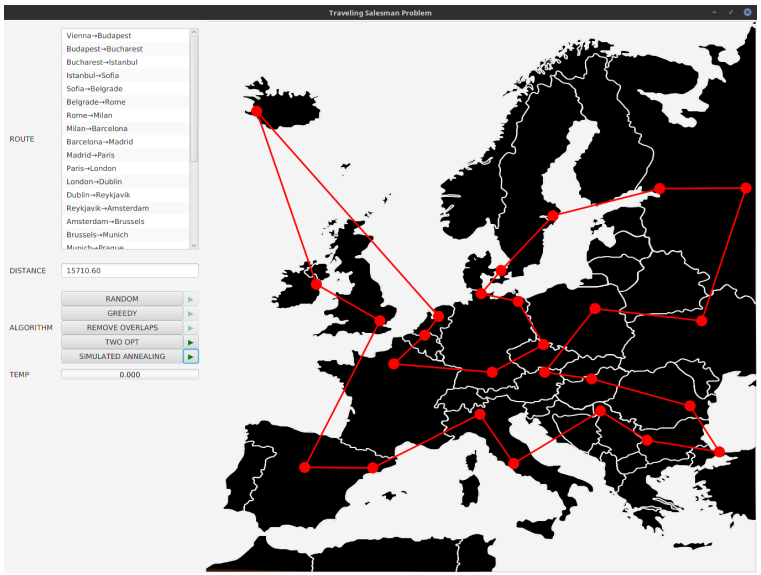
If the priests were able to move disks at a rate of one per second, using the smallest number of moves it would take them $2^{64} - 1$ seconds or roughly 585 billion years to finish, which is about 42 times the current age of the Universe.

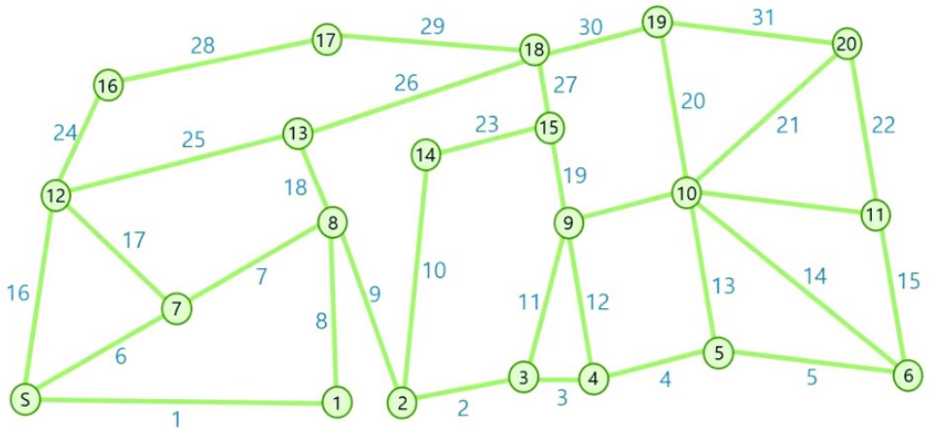
World will end by the time the game ends according to Brahma !

Travelling Salesman Problem



Travelling Salesman Problem





Similarity between hard problems

Is there any connection between the *Partition* problem and *TSP* ?

Similarity between hard problems

Is there any connection between the *Partition* problem and *TSP* ?

NP Completeness Theory

If *Partition* has a fast solution, so does *TSP* and vice versa.

Similarity between hard problems

Is there any connection between the *Partition* problem and *TSP* ?

NP Completeness Theory

If *Partition* has a fast solution, so does *TSP* and vice versa.

Approximation algorithms

If a problem is hard but important to solve, we try to devise faster strategies/heuristics which may not be exact, but provides guarantees on its proximity to the actual best solution.

Which is faster: humans or machines ?

Which is faster: humans or machines ?

For certain things humans seem to do better, namely cognition but ..

Which is faster: humans or machines ?

For certain things humans seem to do better, namely cognition but ..

Computers are much faster when there is a well developed algorithm.

Which is faster: humans or machines ?

For certain things humans seem to do better, namely cognition but ..

Computers are much faster when there is a well developed algorithm.

Machine Learning and AI has been more popular to fill this gap using *data driven* techniques.

Which is faster: humans or machines ?

For certain things humans seem to do better, namely cognition but ..

Computers are much faster when there is a well developed algorithm.

Machine Learning and AI has been more popular to fill this gap using *data driven* techniques.

Caveat

Empirical - no provable guarantees and may lead to unexpected failures.

Which is faster: humans or machines ?

For certain things humans seem to do better, namely cognition but ..

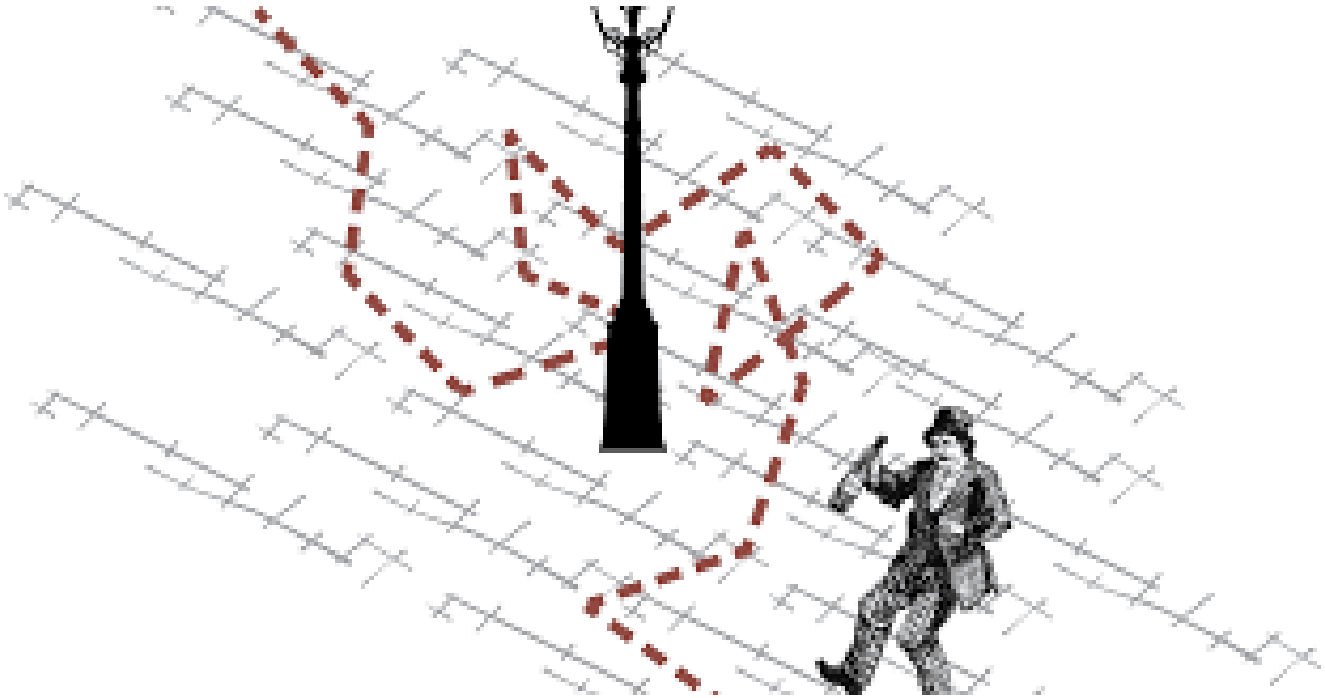
Computers are much faster when there is a well developed algorithm.

Machine Learning and AI has been more popular to fill this gap using *data driven* techniques.

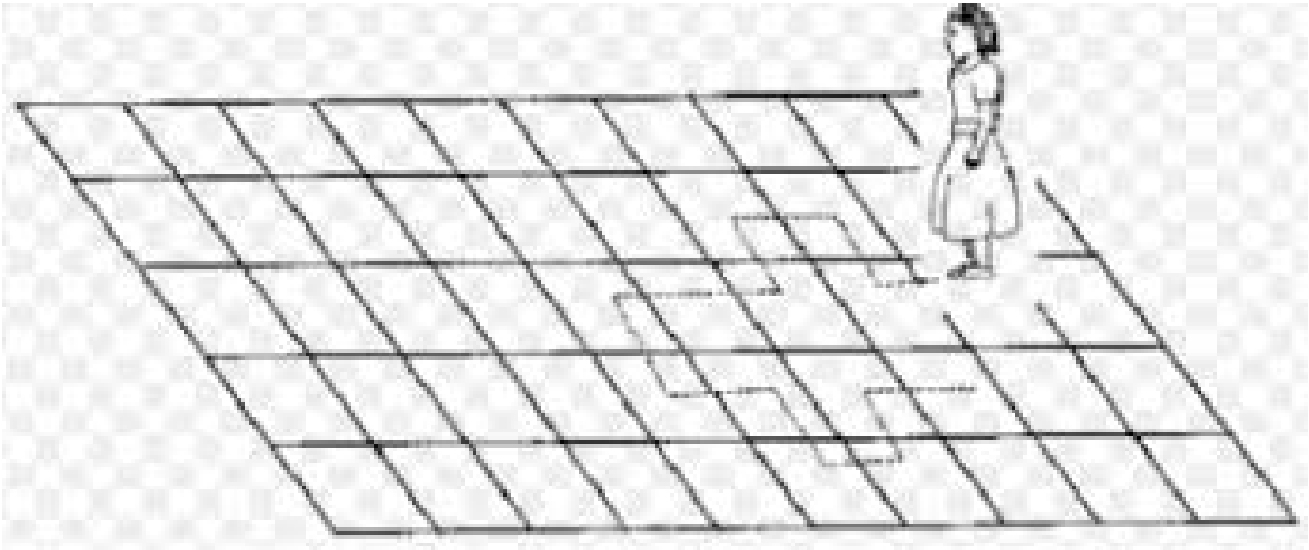
Caveat

Empirical - no provable guarantees and may lead to unexpected failures.
Theory is needed but lagging

When you can't control the outcome

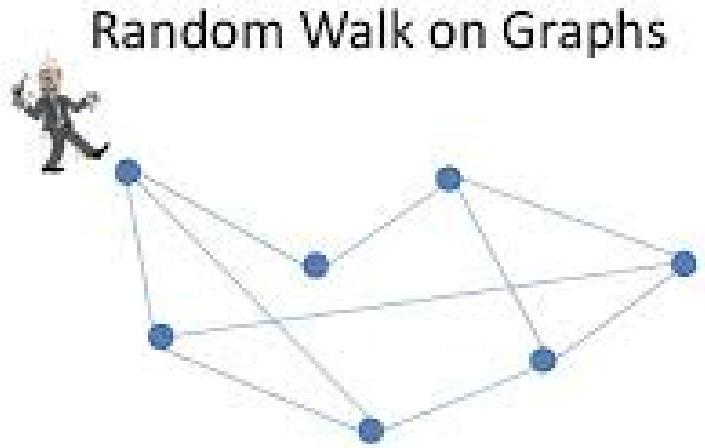


When there are few choices



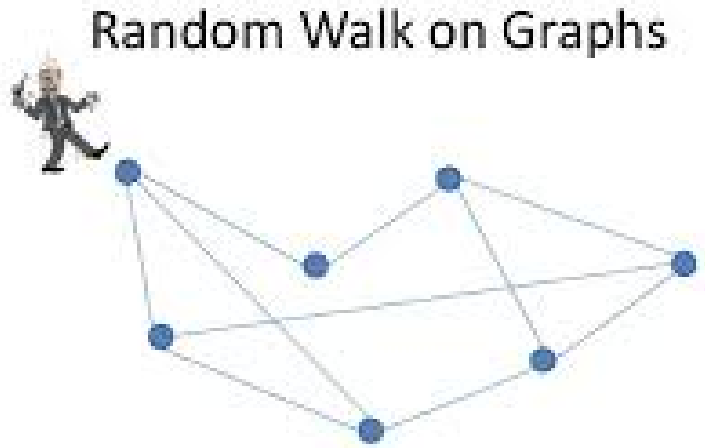
Rat in a maze

Random walk works !



V = number of vertices

Random walk works !



V = number of vertices

Exploring without Maps

By choosing one of the outgoing edges at random from the current vertex, the traveller can visit any destination within $2V^3$ steps (expected).

Exchanging secrets without privacy

Sending secret message is a basic need during love and war.

Exchanging secrets without privacy

Sending secret message is a basic need during love and war.

Codes and ciphers have been used for thousands of years so that messages cannot be understood even if the messenger is compromised or the message is intercepted.

Exchanging secrets without privacy

Sending secret message is a basic need during love and war.

Codes and ciphers have been used for thousands of years so that messages cannot be understood even if the messenger is compromised or the message is intercepted.

But there must be a shared code-book or a previously agreed upon cipher.

Exchanging secrets without privacy

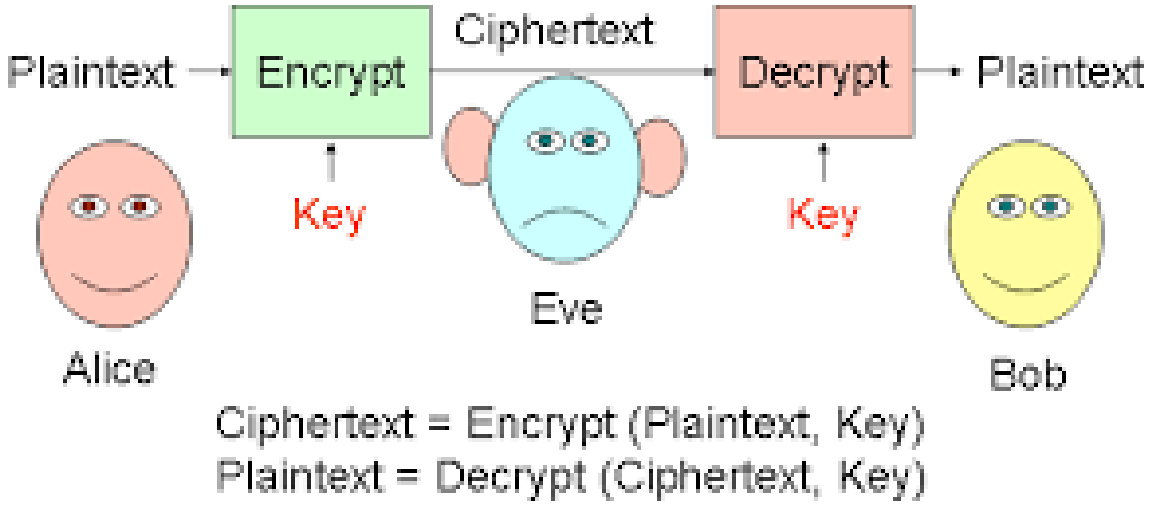
Sending secret message is a basic need during love and war.

Codes and ciphers have been used for thousands of years so that messages cannot be understood even if the messenger is compromised or the message is intercepted.

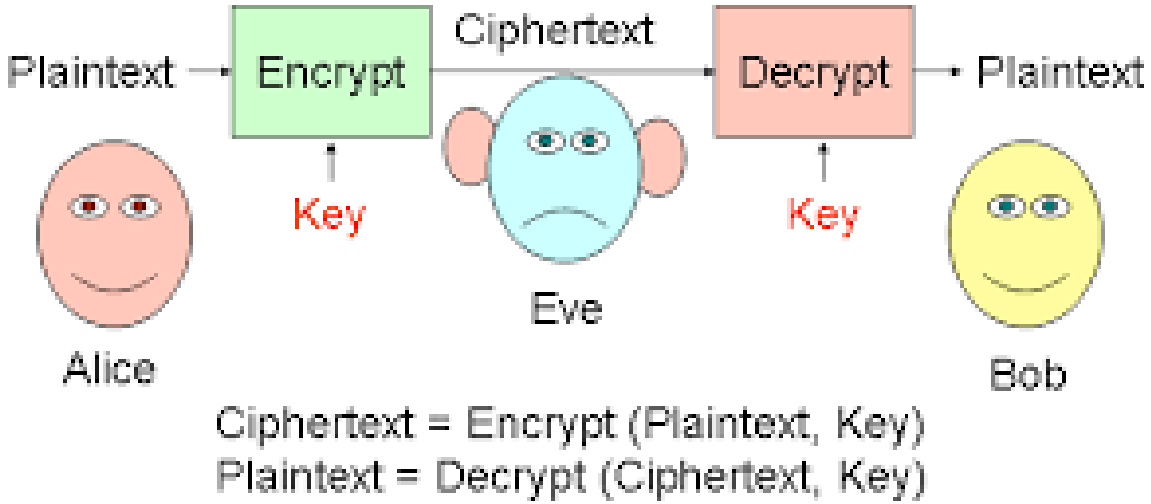
But there must be a shared code-book or a previously agreed upon cipher.

How do you share the code-book or the cipher design without sending this as another message !!

Setting up a cipher in the presence of Eve(sdropper)

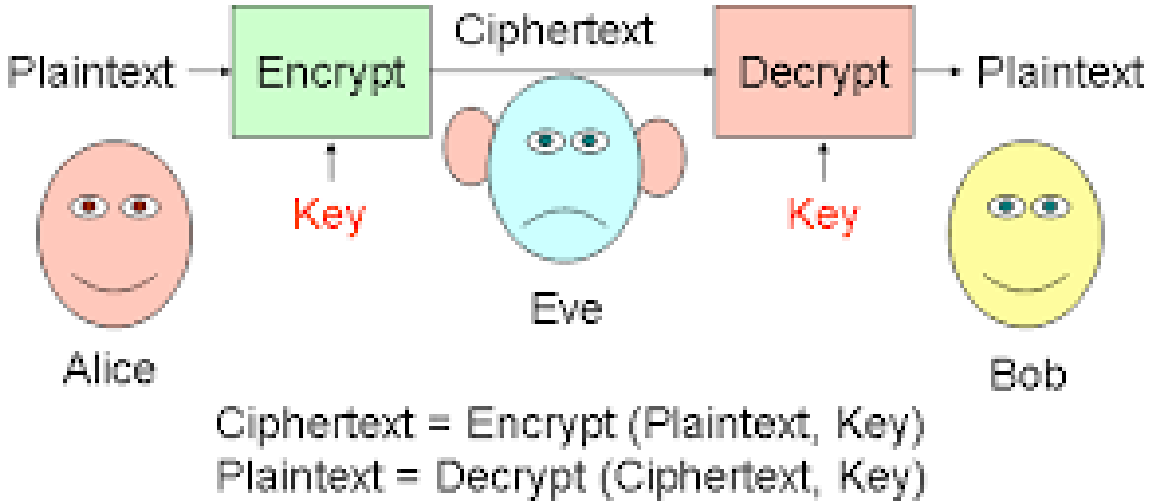


Setting up a cipher in the presence of Eve(sdropper)



Key = (public-key , private-key) for each party.

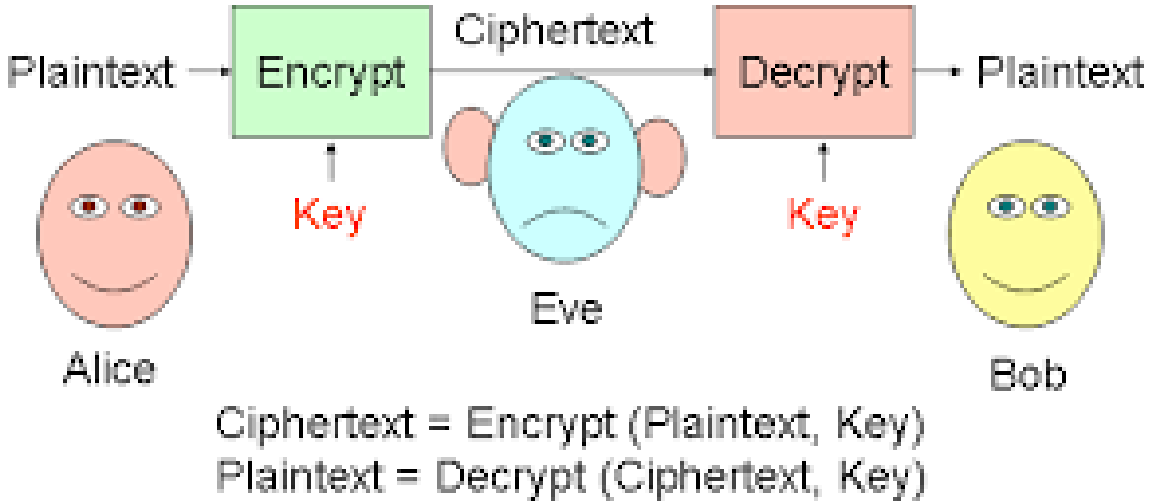
Setting up a cipher in the presence of Eve(sdropper)



Key = (public-key , private-key) for each party.

Alice $\Rightarrow$ Bob : Ciphertext = Encryption (plaintext, Bob's public-key)

Setting up a cipher in the presence of Eve(sdropper)



Key = (public-key , private-key) for each party.

Alice $\Rightarrow$ Bob : Ciphertext = Encryption (plaintext, Bob's public-key)

Bob recovers : Plaintext = Decryption (Message, Bob's private-key)

A related problem

A related problem

Bob gets a message "Let us meet at the coffee shop at 4pm" - Alice

A related problem

Bob gets a message "Let us meet at the coffee shop at 4pm" - Alice

Bob is excited and reaches the coffee shop but after an hour of waiting, calls Alice and she denies having sent any message.

A related problem

Bob gets a message "Let us meet at the coffee shop at 4pm" - Alice

Bob is excited and reaches the coffee shop but after an hour of waiting, calls Alice and she denies having sent any message.

How can this be prevented ?

A related problem

Bob gets a message "Let us meet at the coffee shop at 4pm" - Alice

Bob is excited and reaches the coffee shop but after an hour of waiting, calls Alice and she denies having sent any message.

How can this be prevented ?

Can Alice send **signed** messages ?

A related problem

Bob gets a message "Let us meet at the coffee shop at 4pm" - Alice

Bob is excited and reaches the coffee shop but after an hour of waiting, calls Alice and she denies having sent any message.

How can this be prevented ?

Can Alice send **signed** messages ?

Authentication/Non repudiation using digital signatures

Leveraging Hardness for Encrypt/Decrypt function

Factorization

Finding the factors of a given integer in *believed* to be Hard.
Consider 642536755642980453614572587689083524358756218253420 1
(52 digits)

Leveraging Hardness for Encrypt/Decrypt function

Factorization

Finding the factors of a given integer in *believed* to be Hard.
Consider 642536755642980453614572587689083524358756218253420 1
(52 digits)

Asymmetry in Cryptography

Encryption shoud be easy/efficient
Decryption should be very hard.

Leveraging Hardness for Encrypt/Decrypt function

Factorization

Finding the factors of a given integer in *believed* to be Hard.
Consider 6425367556429804536145725876890835243587562182534201 (52 digits)

Asymmetry in Cryptography

Encryption shoud be easy/efficient
Decryption should be very hard.
Public Key Cryptography
Diffie-Hellman 1976, Rivest-Shamir-Adleman (RSA) 1978

Asymmetry : Verification by Multiplying is easy but factorization doesn't have an efficient algorithm unless ..

Leveraging Hardness for Encrypt/Decrypt function

Factorization

Finding the factors of a given integer in *believed* to be Hard.
Consider 6425367556429804536145725876890835243587562182534201 (52 digits)

Asymmetry in Cryptography

Encryption shoud be easy/efficient
Decryption should be very hard.
Public Key Cryptography
Diffie-Hellman 1976, Rivest-Shamir-Adleman (RSA) 1978

Asymmetry : Verification by Multiplying is easy but factorization doesn't have an efficient algorithm unless ..
You can build a reliable large quantum computer

Building Faster Computers

Time and work problems

If 30 people can cut the crop of a 10 acre farm in 15 days, how many work days is needed to cut the crop of a 5 acre field using 20 people ?

Building Faster Computers

Time and work problems

If 30 people can cut the crop of a 10 acre farm in 15 days, how many work days is needed to cut the crop of a 5 acre field using 20 people ?

Is this true for all activities ?

Building Faster Computers

Time and work problems

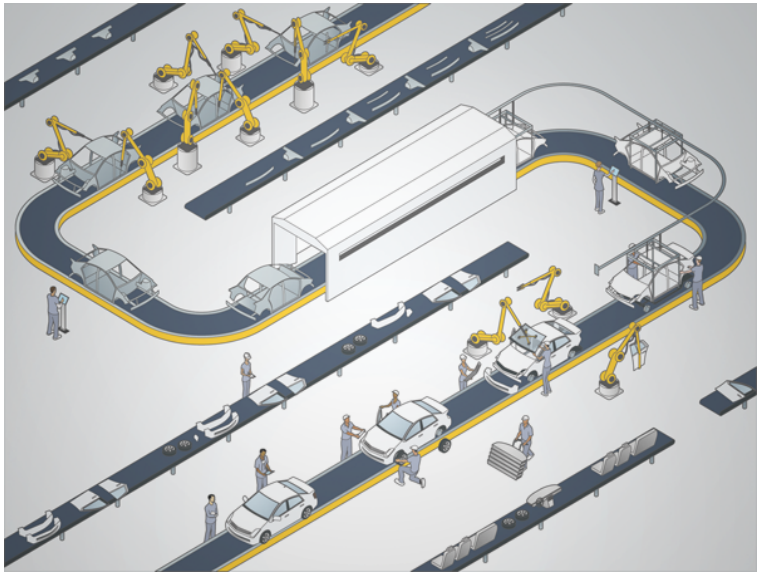
If 30 people can cut the crop of a 10 acre farm in 15 days, how many work days is needed to cut the crop of a 5 acre field using 20 people ?

Is this true for all activities ?

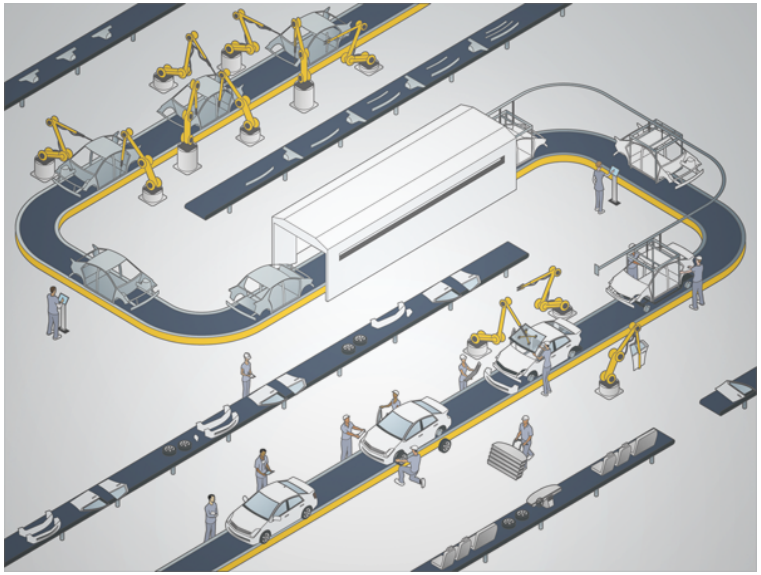
Making Tea

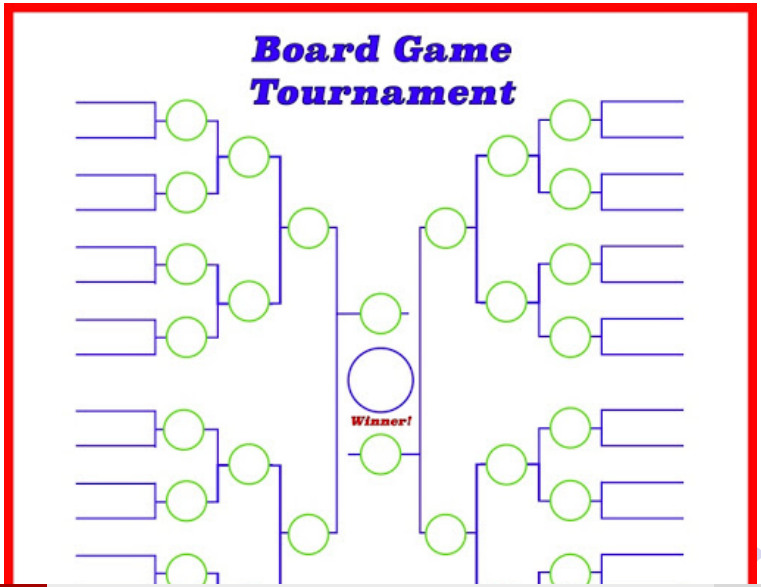
1. Boil water
2. Add Tea leaves
3. Add Milk
4. Add Sugar
5. Pour and drink

Examples of other Activities

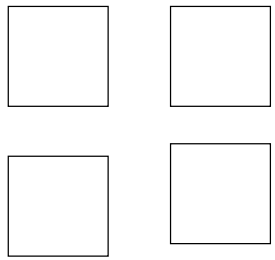


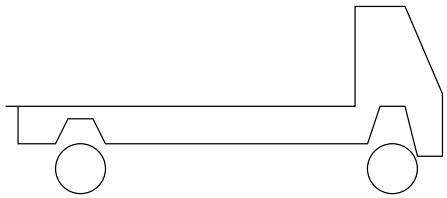
Examples of other Activities



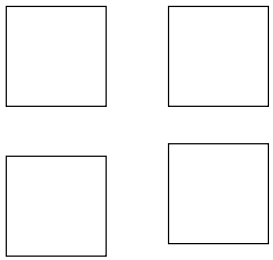


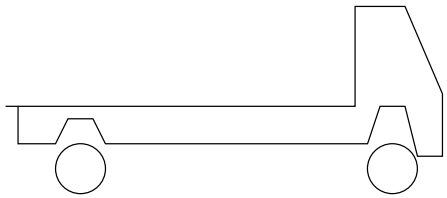
Easy parallelism





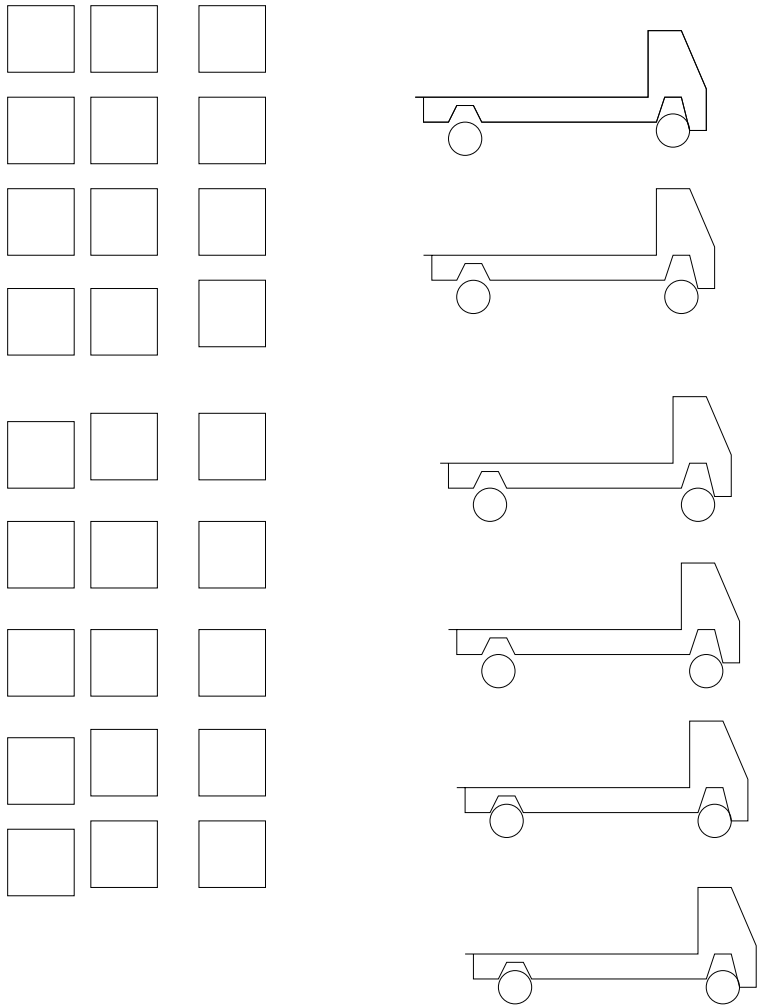
Easy parallelism





More containers, more trips , more time

Easy parallelism



More challenging

ADD

25346437893562435635478598743243593546725363546980508785132036 7

63583733830863024353997265340383616353768935408516253497465093 6

Elementary algorithm : Add and Carry from left to right

More challenging

ADD

25346437893562435635478598743243593546725363546980508785132036 7

63583733830863024353997265340383616353768935408516253497465093 6

Elementary algorithm : Add and Carry from left to right
Number of steps = Number of digits

More challenging

ADD

25346437893562435635478598743243593546725363546980508785132036 7

63583733830863024353997265340383616353768935408516253497465093 6

Elementary algorithm : Add and Carry from left to right
Number of steps = Number of digits

Multiple processors

CAN RAVAN DO IT FASTER USING MULTIPLE HEADS ??
Multicore/CUDA

More challenging

ADD

25346437893562435635478598743243593546725363546980508785132036763583733830863024353997265340383616353768935408516253497465093635837338308630243539972653403836163537689354085162534974650936

Elementary algorithm : Add and Carry from left to right
Number of steps = Number of digits

Multiple processors

CAN RAVAN DO IT FASTER USING MULTIPLE HEADS ??
Multicore/CUDA

One can think about CLOUD as a massive heterogeneous parallel processing environment that is scheduling resources optimally to many jobs running simultaneously.

Where should I do PhD in TCS (Track A FSTTCS)

Where should I do PhD in TCS (Track A FSTTCS)

Some of the most visible active groups in Algorithms and Complexity

- Algorithms : IIT Delhi , IIT Bombay, IISc , IMSc, ISI Kolkata
- Complexity : IIT Kanpur, IMSc, IISc, TIFR, CMI
- Cryptography, Security : IIT Bombay, IIT Madras, IISc, IIT Kanpur, ISI
- Distributed and Parallel Algorithms: IIIT Hyderabad, IISc
- Quantum Computation : TIFR, IIIT Delhi

Where should I do PhD in TCS (Track A FSTTCS)

Some of the most visible active groups in Algorithms and Complexity

- Algorithms : IIT Delhi , IIT Bombay, IISc , IMSc, ISI Kolkata
- Complexity : IIT Kanpur, IMSc, IISc, TIFR, CMI
- Cryptography, Security : IIT Bombay, IIT Madras, IISc, IIT Kanpur, ISI
- Distributed and Parallel Algorithms: IIIT Hyderabad, IISc
- Quantum Computation : TIFR, IIIT Delhi

More options

There are smaller active groups and talented individuals in most IITs and other universities - that one can explore by visiting websites as per indvidual interests.