MCA302

Theory of Computation and Compilers

Ann Baby, Ph.D.
9895012317
ann@rajagiri.edu

Theory of Computation (TOC)

- A branch of Computer Science that is concerned with how problems can be solved using algorithms and how efficiently they can be solved

- Helps to understand how people have thought of Computer Science in the past years

Importance of Theory of computation

- TOC forms the basis for:
  - Writing efficient algorithms that run in computing devices.
  - Programming language research and their development.
  - Efficient compiler design and construction.

Theory Of Computation

- Beyond
  - Writing Code
  - Compiling Code
  - Error Correction

  - What are the capabilities and limitations of computers ??

Branches of TOC

- Automata Theory.
- Computability Theory.
- Complexity Theory.

Key considerations of Computational Problems

- What can and cannot be computed.
- Mathematically evaluate
  - Speed of such computations.
  - The amount of memory in use during such computations.

Machine Designing Example -1

- Accepts all binary strings ending in 0

Machine Designing Example -2

- Accepts all valid Java Code
- Java Code → Binary Equivalent → Check validity

- Compilers → TOC

Machine Designing Example -3

- Accepts all valid Java Code and never goes into an infinite loop
- Java Code → Binary Equivalent → Check validity
- Wrong output
- No output

Mathematical Preliminaries

- Sets

  x is an element of a set S , $x \in S$.

  x is not an element of a set S , $x \notin S$.

  $S = \{0, 1, 2\}$.

$$S = \{i : i > 0, i \text{ is even}\}$$

Basic Set Operations

$$S_1 \cup S_2 = \{ \ldots x \in S_2\},$$

$$\overline{S} = \{x : x \in U, x \notin S\}.$$

$$S_1 \cap S_2 = \{x : x \in S_1 \text{ and } x \in S_2\},$$

$$S_1 - S_2 = \ldots \text{d } x \notin S_2\}.$$

$$\overline{S_1 \cup S_2} = \overline{S}_1 \cap \overline{S}_2,$$

$$\overline{S_1 \cap S_2} = \overline{S}_1 \cup \overline{S}_2,$$

$$S = S_1 \times S_2 = \{(x, y) : x \in S_1, y \in S_2\}.$$

Find the Cartesian product of A X B

- A = {1,2,3}
- B = {x,y}

Functions and Relations

- A function is a rule that assigns to elements of one set (the function domain) a unique element of another set (the range).

  $f : S1 \rightarrow S2$

  $Domain\ f = \{x \in S1 \mid (x, y) \in f, \text{ for some } y \in S2\} = Df$

  $Range\ f = \{y \in S2 \mid (x, y) \in f, \text{ for some } x \in S1\} = Rf$

- R**elation** is defined as the collection of ordered pairs, which contains an object from one set to the other set.

  - An equivalence relation on a set A is a reflexive, symmetric, and transitive relation.

- **Reflexive**: A relation is said to be reflexive, if (a, a) ∈ R, for every a ∈ A.
- **Symmetric**: A relation is said to be symmetric, if (a, b) ∈ R, then (b, a) ∈ R.
- **Transitive**: A relation is said to be transitive if (a, b) ∈ R and (b, c) ∈ R, then (a, c) ∈ R.

For R = {1,2,3}, find whether the following relation is equivalent.

- 1. R1= { (1,1), (2,2) , (3,3) }

For R = {1,2,3}, find whether the following relation is equivalent.

- R2= { (1,1), (2,2) , (3,3) , (3,2) , (1,3) }

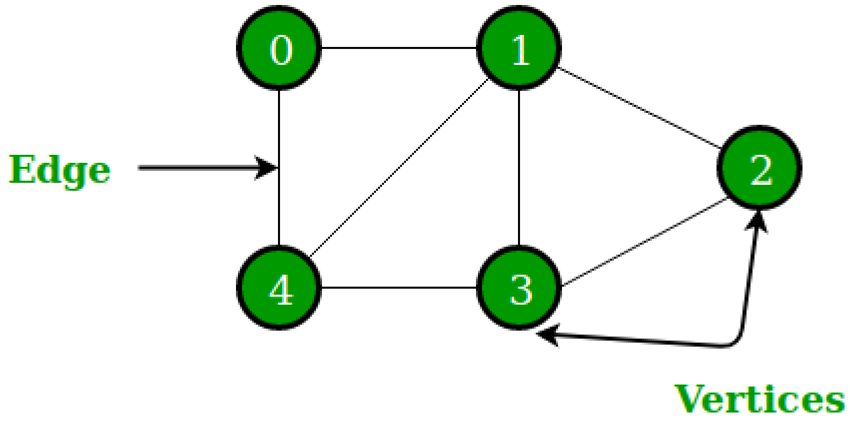
For R = {1,2,3}, find whether the following relation is equivalent.

- R3= { (1,1), (2,2) , (3,3) , (2,1) , (1,2) }

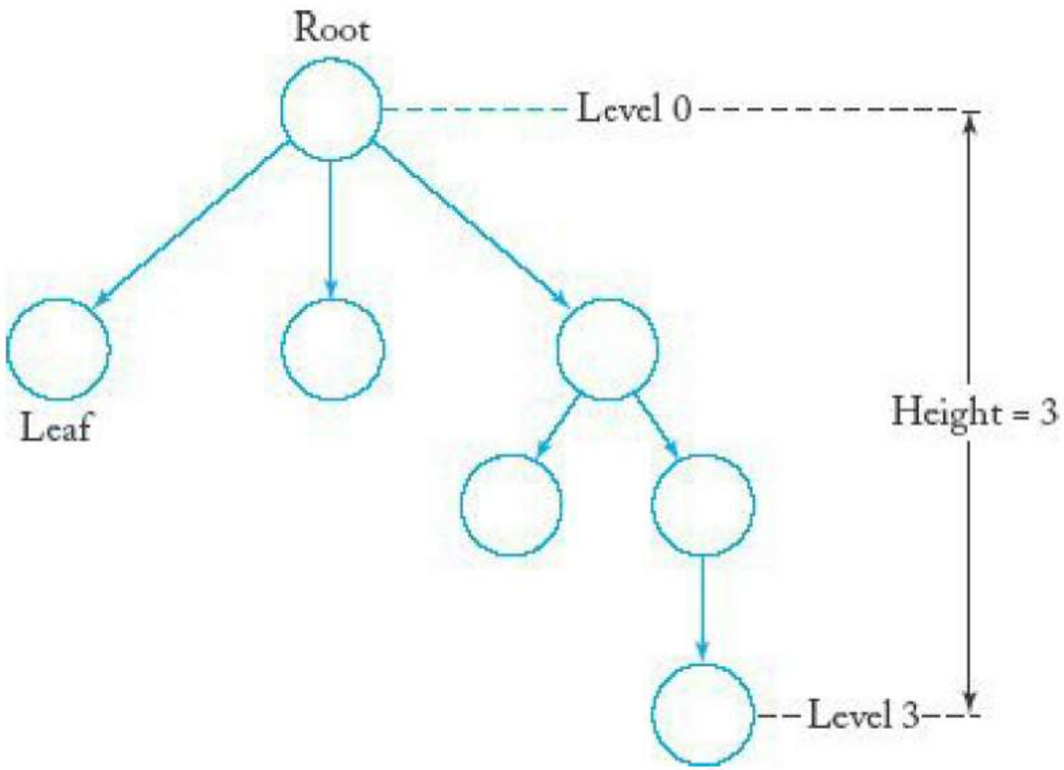

Mistakes
are proof
that you
are trying.

Graphs

- $G = \{\{V_1, V_2, V_3, V_4, V_5, V_6\}, \{E_1, E_2, E_3, E_4, E_5, E_6, E_7\}\}$



Trees



Proof Techniques

- Proof by Induction
- **Proof by Contradiction**

Proof by Induction

- We have a sequence of statements P1, P2, · · · , about which we want to make some claim.
- Suppose that we know that the claim holds for all statements P1, P2, · · · , up to Pn.
- We then try to argue that this implies that the claim also holds for Pn+1.

- If we can carry out this inductive step for all positive n, and if we have some starting point for the induction, we can say that the claim holds for all statements in the sequence.

Steps in Proof by Induction

- The starting point for an induction is called the basis.
- The assumption that the claim holds for statements P1, P2, · · · , Pn is the induction hypothesis.
- The argument connecting the induction hypothesis to Pn+1 is the induction step.
- Conclusion

- Prove by mathematical induction

  n(**n**+1) / 2, for **n** a **natural number**.

- Prove by mathematical induction

$$\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Proof by Contradiction

- Suppose we want to prove that a statement P is true,
  - First, lets assume P is false.
  - Wait and see what this assumption leads us to .
  - If we arrive at a conclusion which we know is incorrect, put the blame on the assumption.

  - Thus, we can say our assumption is false.
  - Therefore, P is true.

Prove that $\sqrt{2}$ is irrational (Use contradiction)

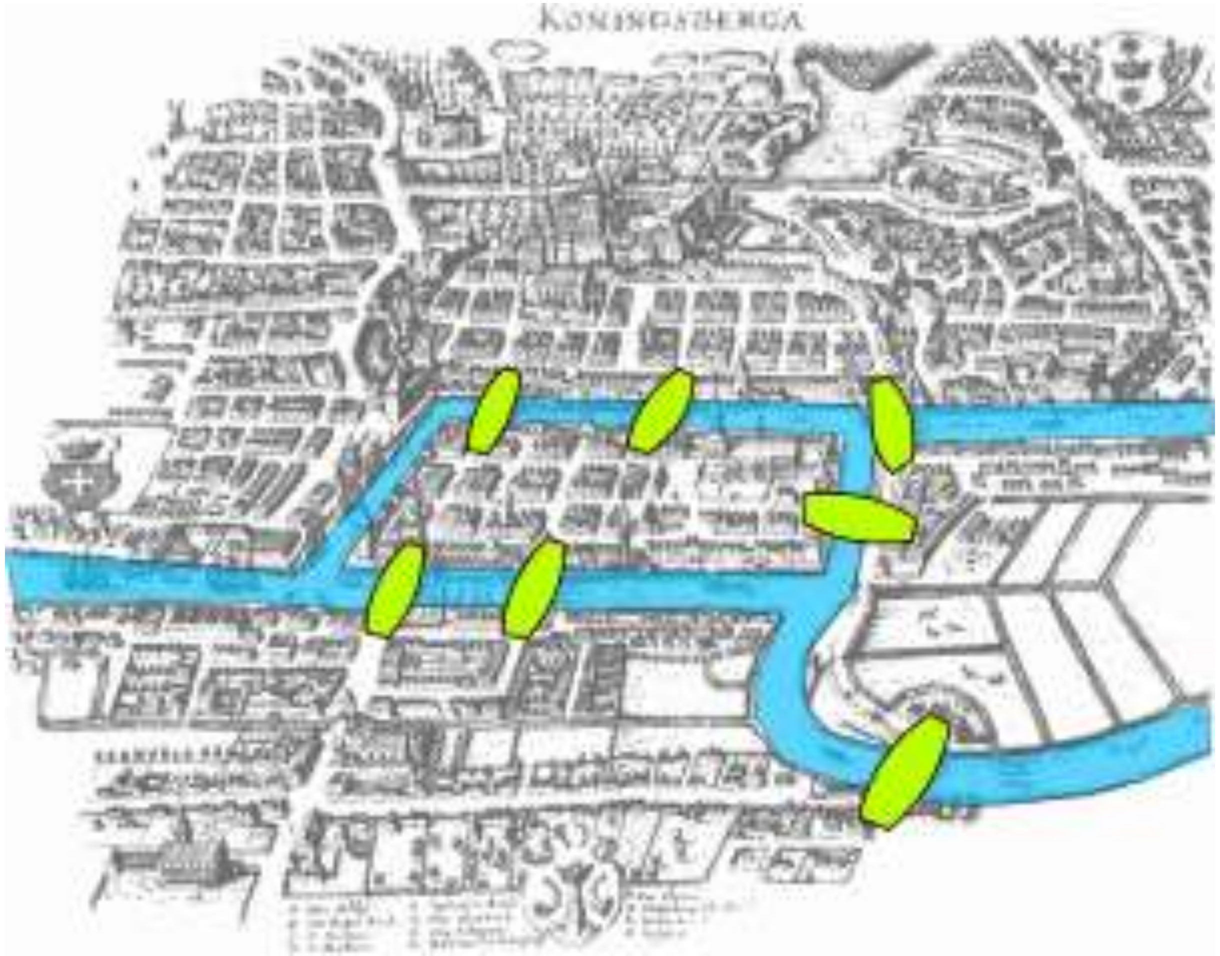
Problems in Computation

- Solvable
  - Exists a solution
  - Exists an algorithm
  - Eg: 2+ 2
- Unsolvable
  - Very complex
  - Divide by 0
  - Time travelling
  - Disappear at the clap of my hand

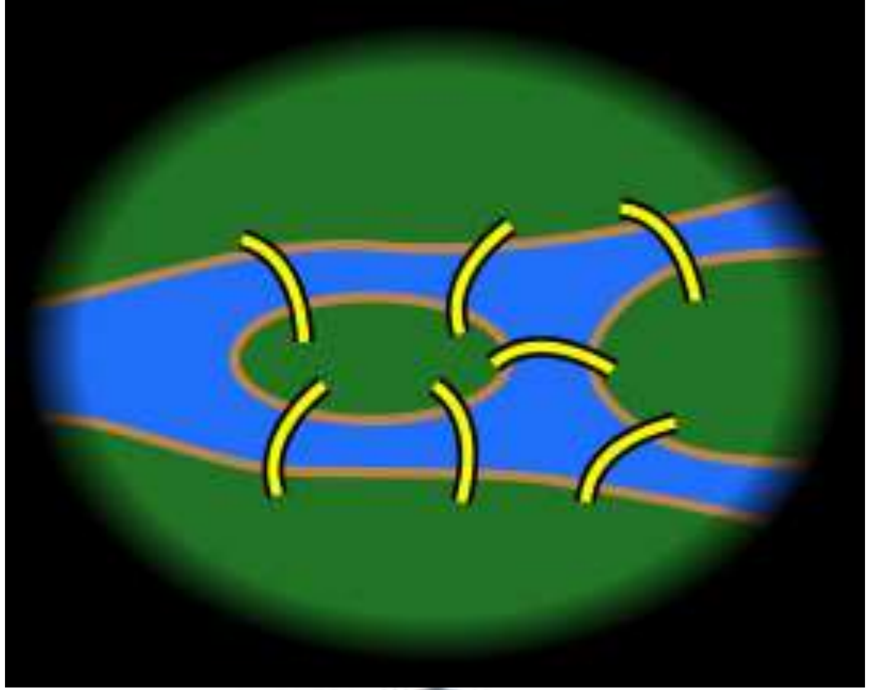
How a Russian city gave birth to the graph theory!!!

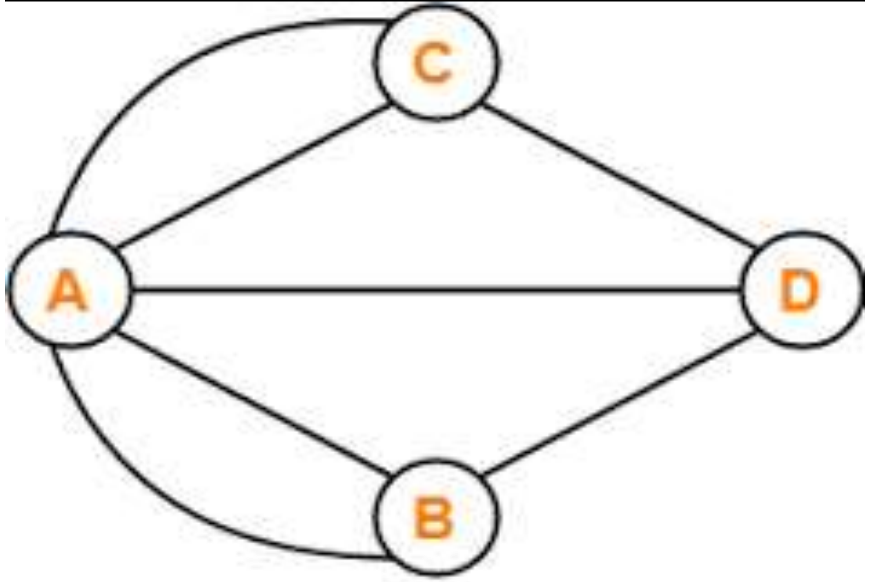
- Konigsberg

Konigsbergs Problem



Euler’s Representation





Graph Representation

Problems in Computer Science

Problems

Solvable

Algorithm / Procedure

Prove that there exists no solution

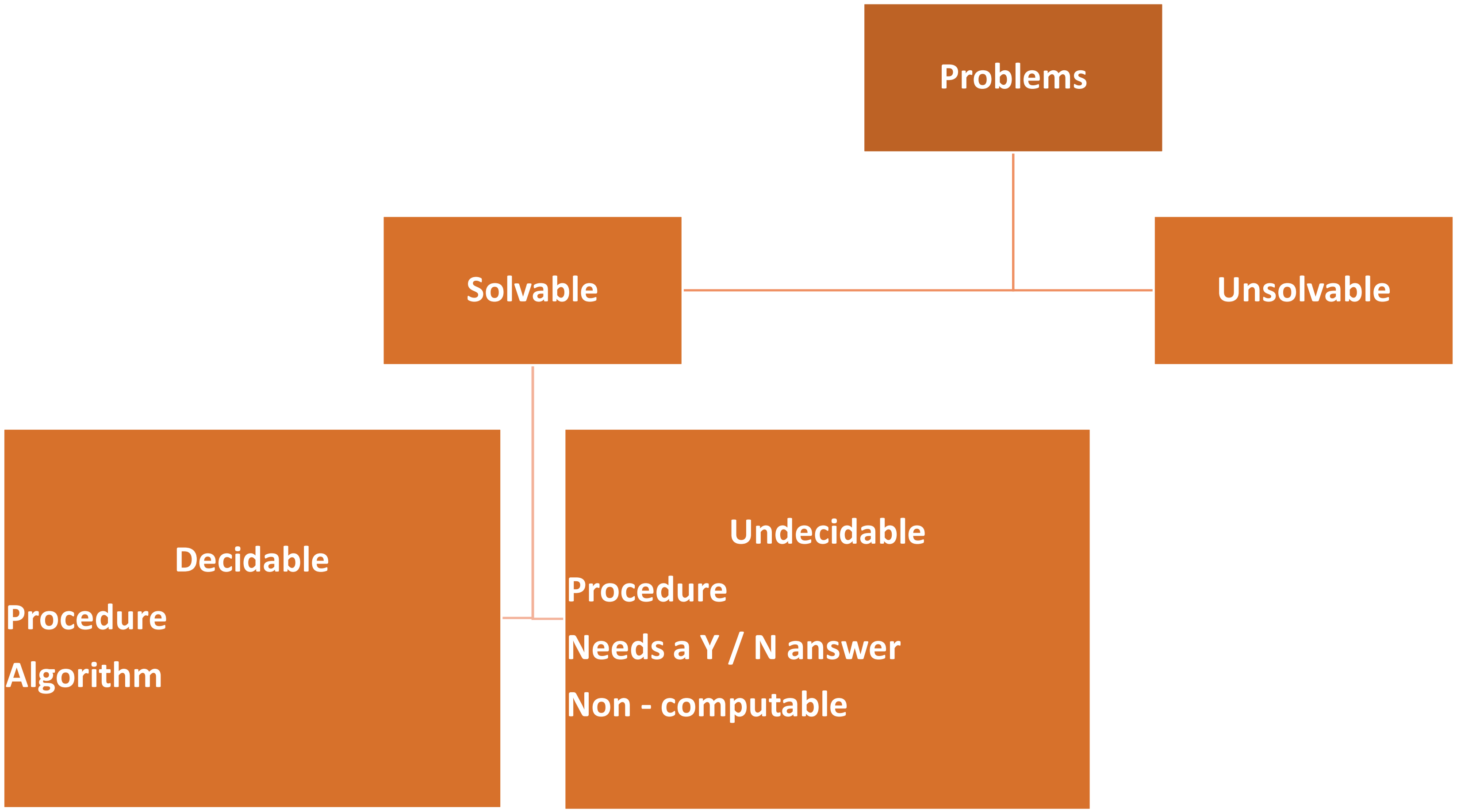
Unsolvable

Temporary Status of a problem

Wrt time

Unable to say whether it can be solved/not solved

Problems in Computer Science



Undecidable/ Non Computational Problems

- Halting Problem
- Post Correspondence Problem
- Wang Tiles
- Magic : The Gathering (Game)
- etc

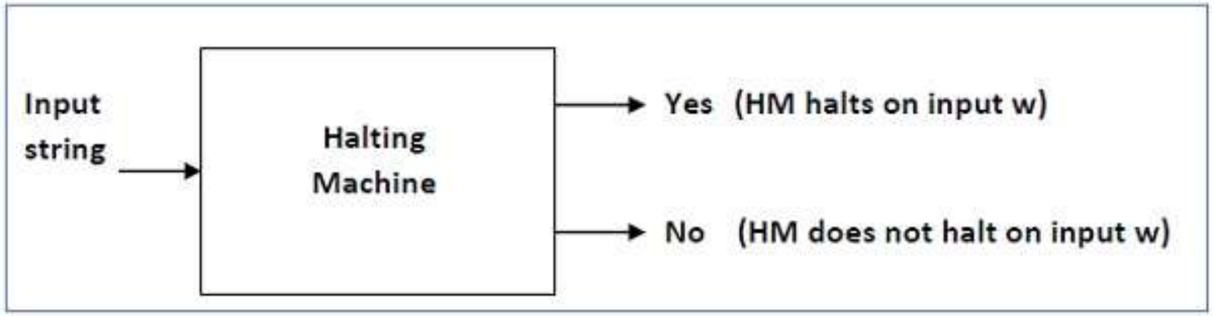
Halting Problem

- **Halting** means that the program on certain input
  - Halt
    - will accept it and **halt**
    - reject it and **halt**
  - Not Halt
    - it would never go into an infinite loop.

  - It is not a problem, but it asks a question.
  - Given a program, will it halt ???

Block Diagram for Halting Problem

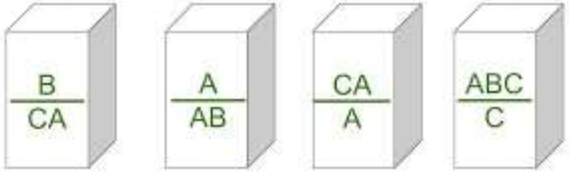
- Is it possible to tell whether a given machine will halt for some given input, w
- *Entscheidungs problem*



- *Can you design a generalized algorithm which will take in a program as input, and tell whether it will halt or not ?*

Post Correspondence Problem (PCP)

- Introduced by Emil Post in 1946
- Arranging of Dominos
- Numerator – top
- Denominator – bottom



- “Find a **finite sequence** such that the top and bottom are the same”

- Arrange however you want
- Repetitions possible
- **FINITE SEQUENCE**

2. Arrange the following list

A - num	B -den
1	111
10111	10
10	0

POST CORRESPONDENCE PROBLEM

1. Arrange the following Dominos

1	2	3	4
B	A	CA	ABC
CA	AB	A	C

3. Arrange the following list

A -num	B - den
10	101
011	11
101	011

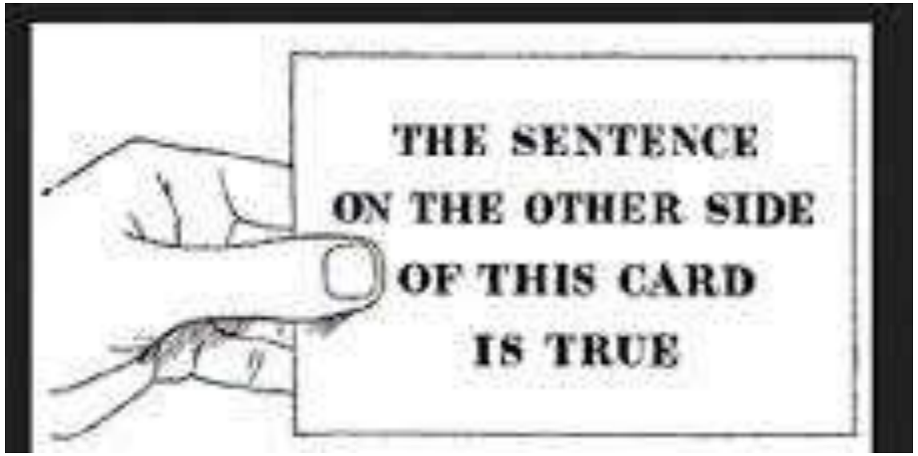
Paradox

- A logically self-contradictory statement
- A statement in which it seems that it is both true and untrue at the same time
- Antinomy

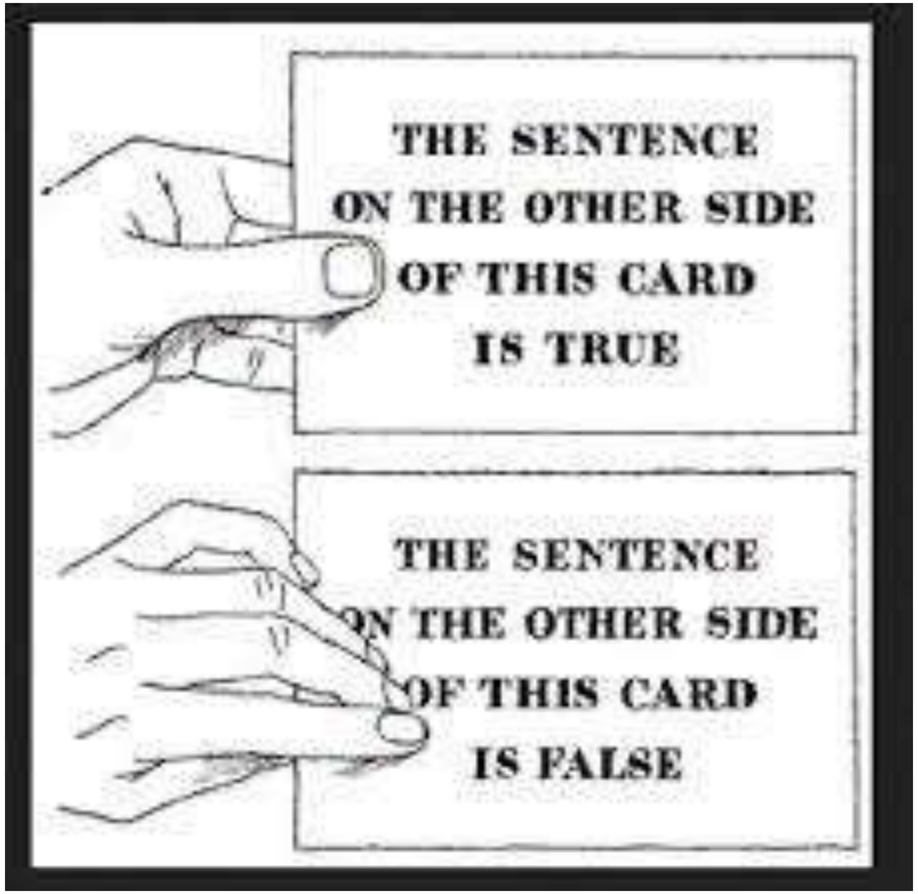
Examples

- This is the beginning of the end

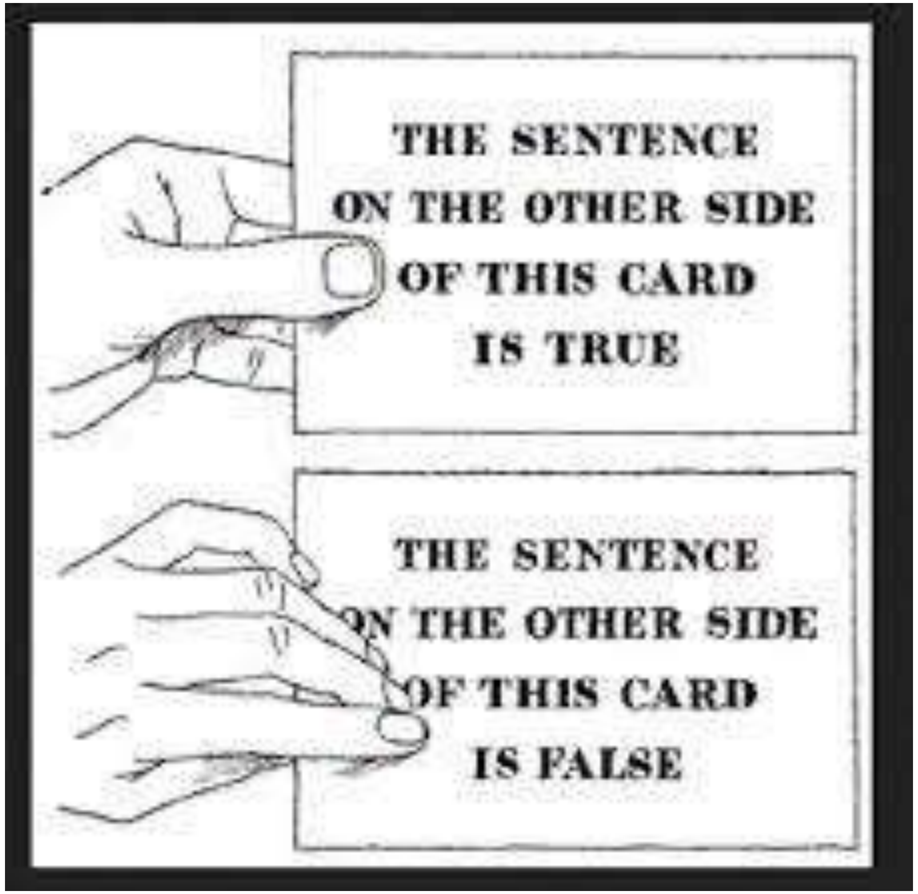
The Card paradox / The Liar’s Paradox

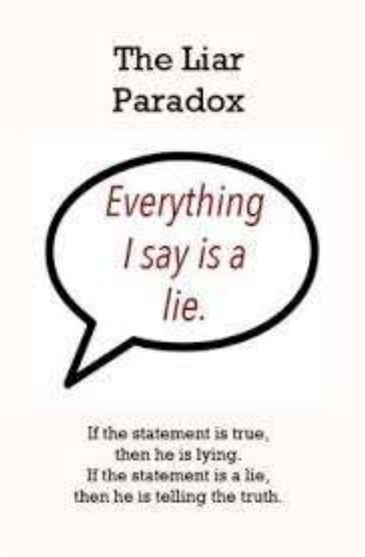


The Card paradox / The Liar's Paradox



The Card paradox / The Liar's Paradox





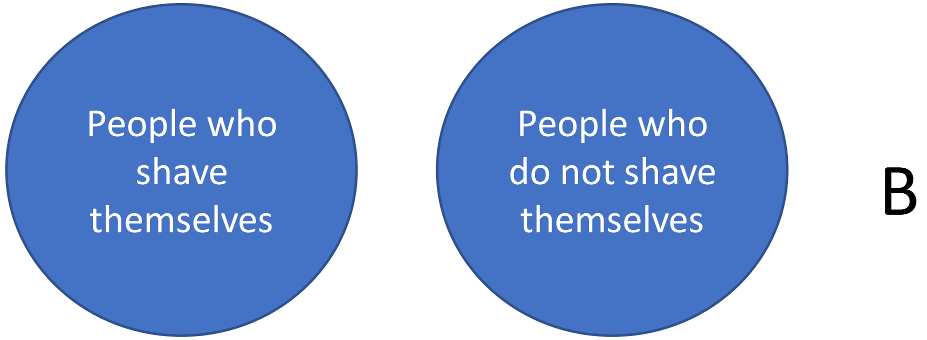
The Barbers Paradox

- Once upon a time there was a small village, and in this village lived a barber
- Barber shaved all the villagers who did not shave themselves.
- Barber shaved none of the villagers who did shave themselves.

- The question :

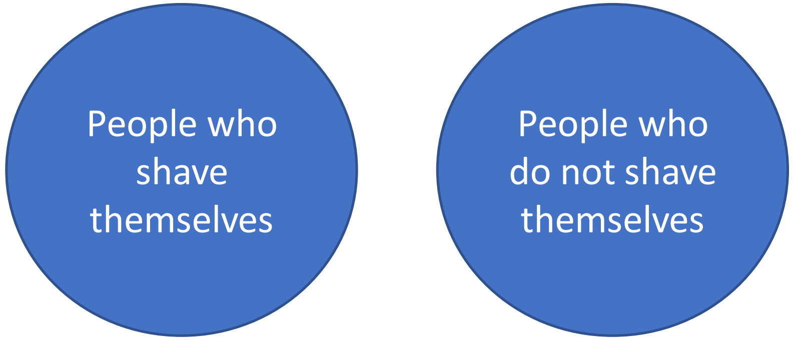
> The barber is a man in town who shaves all those, and only those, men in town who do not shave themselves.
>
> Does the barber shave himself?

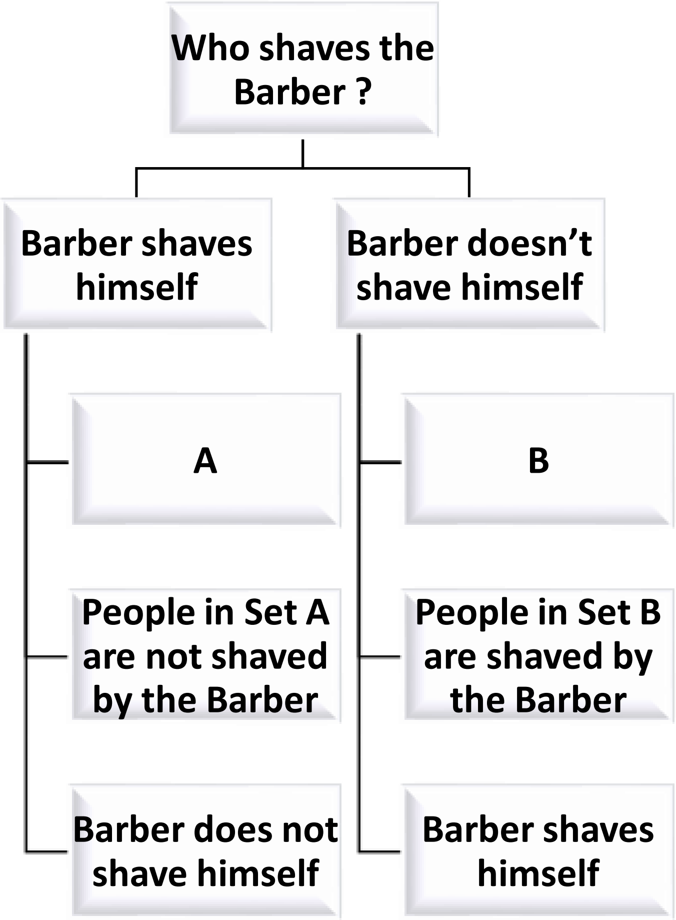
Does the barber shave himself?



Does the barber shave himself?

- A B





Naïve Set Theory – Comprehension Principle

- Universal set U ={x | x is a set }
- It means U Є U

- Eg:

- Think of an imaginary bag stuffed with many balls.
- This means the bag U is stuffed inside itself
- Really Weird !!!

- A = { 1, 4, 6, 9}
- 1 ∈ A , … 9 ∈ A
- But { 1, 4, 6, 9} $\notin$ A

- B = {2, 9 , { 1, 4, 6, 9}, 3 , 1 }
- Here, { 1, 4, 6, 9} ∈ B
- But B $\notin$ B , as {2, 9 , { 1, 4, 6, 9}, 3 , 1 } is not one of the four elements of B

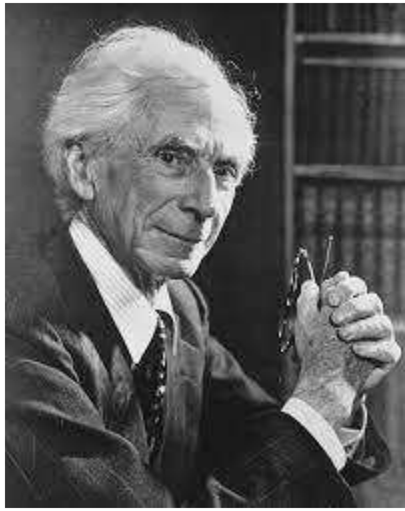
- Thus some sets are elements of themselves and some sets are not …

Russell’s Paradox / Russell’s Antinomy -1901

- Bertrand Russell - Author of ***Principia Mathematica***
- A classic set theoretic paradox to prove the looseness of Naïve Set theory

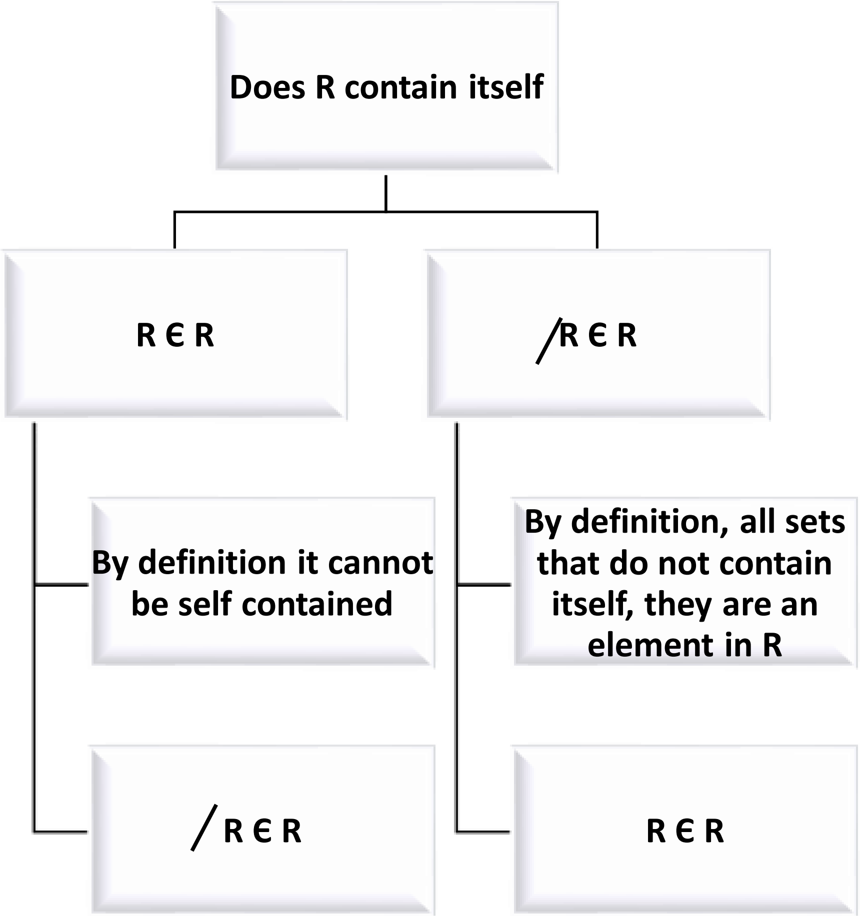
- R is the set of sets which do not contain themselves
- Mathematically , R = $\{S : S \in S\}$

- Does R contain itself ?
- Mathematically, Does R $\in$ R ?



$$R = \{S : S \notin S\}$$

- Neither of them hold !!!
- A paradox indeed ….

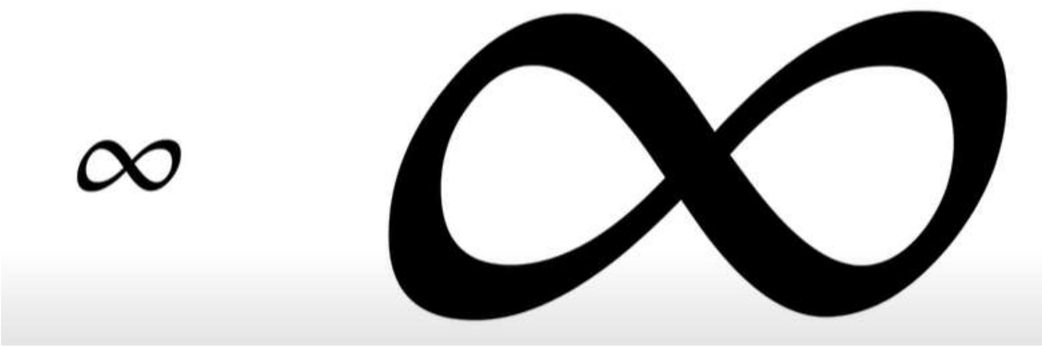


Infinities

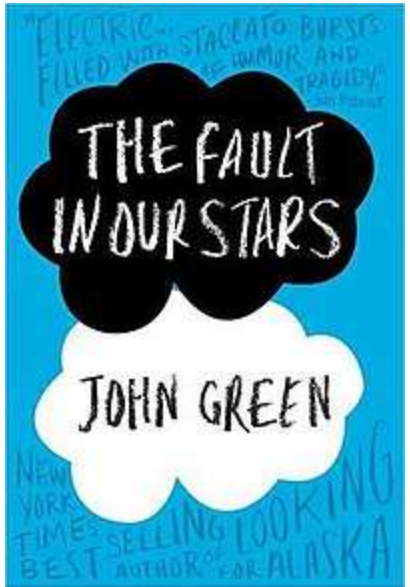
∞

- Boundless or Endless
- The set of Real numbers is infinite
- The set of Natural numbers is infinite


Some
INFINITIES
are
BIGGER
THAN OTHER
Infinities



Courtesy :


THE FAULT IN OUR STARS
JOHN GREEN

A Possible Mapping $\quad N \rightarrow R\ [0.0 - 1.0]$

Chair	Student
0	0.5 2 3 9 5 4 7 1 6 ….
1	
2	
3	
4	
5	
….	

A Possible Mapping $N \rightarrow R$ [0.0 – 1.0]

Chair	Student
0	0.5 2 3 9 5 4 7 1 6 ….
1	0.5 2 4 9 5 4 7 1 6 ….
2	
3	
4	
5	
….	

A Possible Mapping $N \rightarrow R$ [0.0 – 1.0]

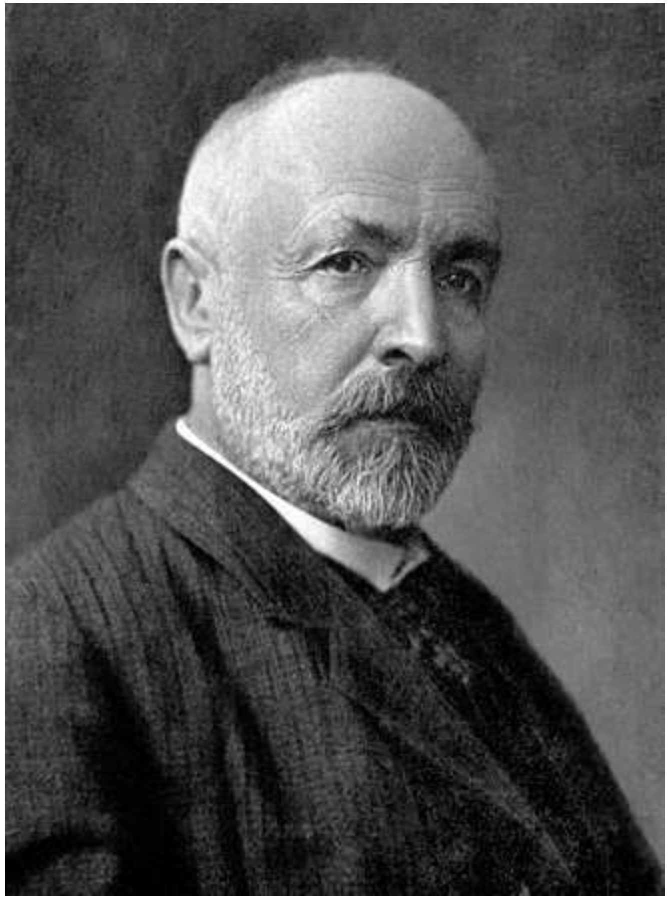
Chair	Student
0	0.5 2 3 9 5 4 7 1 6 ….
1	0.5 2 4 9 5 4 7 1 6 ….
2	0.1 2 3 1 2 3 1 2 3 …
3	0. 9 9 9 9 9 9 9 9 ….
4	0. 0 0 0 0 0 0 0 1…
5	0.3 6 8 9 2 5 8 1 1 …
….	

Chair	Student
0	0.5 2 3 9 5 4 7 1 6 ….
1	0.5 2 4 9 5 4 7 1 6 ….
2	0.1 2 3 1 2 3 1 2 3 …
3	0. 9 9 9 9 9 9 9 9 ….
4	0. 0 0 0 0 0 0 0 1…
5	0.3 6 8 9 2 5 8 1 1 …
….	

- There is always going to be a student without a chair
- There is always going to be a real number which cannot be mapped onto a natural number
- Proved by the Diagonalization Argument

Georg Cantor

- Creator of the Set Theory
- Cantor’s Diagonalization Argument - 1891
- **Cantor’s Infinities**



Georg Cantor’s Diagonalization Argument - 1891

Chair	Student
0	0 5 2 3 9 5 4 7 1 6 ….
1	0.5 2 4 9 5 4 7 1 6 ….
2	0.1 2 3 1 2 3 1 2 3 …
3	0. 9 9 9 9 9 9 9 9 ….
4	0. 0 0 0 0 0 0 0 1…
5	0.3 6 8 9 2 5 8 1 1 …
….	

Cantor’s Diagonalization Argument

Chair	Student
0	0.5 2 3 9 5 4 7 1 6 ….
1	0.5 2 4 9 5 4 7 1 6 ….
2	0.1 2 3 1 2 3 1 2 3 …
3	0. 9 9 9 9 9 9 9 9 ….
4	0. 0 0 0 0 0 0 0 1…
5	0.3 6 8 9 2 5 8 1 1 …
….	
	0.6 3 4 0 1 6 ….

Cantor’s Diagonalization Argument

Chair	Student
0	0 5 2 3 9 5 4 7 1 6 ….
1	0.5 2 4 9 5 4 7 1 6 ….
2	0.1 2 3 1 2 3 1 2 3 …
3	0. 9 9 9 9 9 9 9 9 ….
4	0. 0 0 0 0 0 0 0 1…
5	0.3 6 8 9 2 5 8 1 1 …
….	
	0.6 3 4 0 1 6 ….

No Chair for the New Stud

Cantor’s Diagonalization Argument

Chair	Student
0	0 5 2 3 9 5 4 7 1 6 ....
1	0.5 2 4 9 5 4 7 1 6 ....
2	0.1 2 3 1 2 3 1 2 3 ...
3	0. 9 9 9 9 9 9 9 9 ....
4	0. 0 0 0 0 0 0 0 1...
5	0.3 6 8 9 2 5 8 1 1 ...
....	
	0.6 3 4 0 1 6 ....
	0. 7 4 5 2 2 7 ....

No Chair for the New Stud

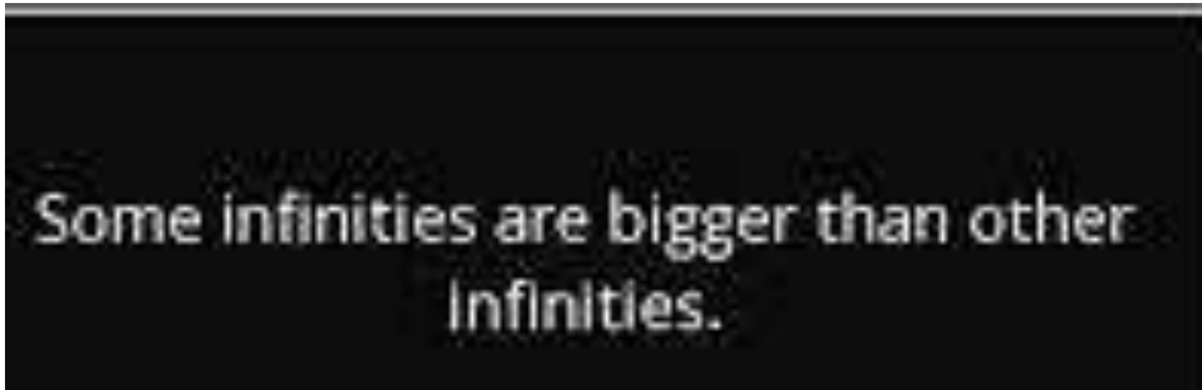
No Chair for the New Stud

N→R [0.0 – 1.0]

- Real numbers are a larger infinity than the infinity of Natural Numbers

- Natural Numbers – Countable infinity
- Real Numbers – Uncountable Infinity

- Thus, Real Numbers are Uncountable



No :	Function, f(n) N→N
0	$f_0(n)$
1	$f_1(n)$
2	$f_2(n)$
3	$f_3(n)$
4	$f_4(n)$
5	$f_5(n)$
….	…

No :	Function, f(n) N→N
0	$f_0(n) = \{(0,0), (1,2), (2,4), (3,6), \ldots\}$
1	$f_1(n) = \{(0,0), (1,1), (2,2), (3,3), \ldots\}$
2	$f_2(n) = \{(0,0), (1,1), (2,4), (3,9), \ldots\}$
3	$f_3(n) = \{(0,0), (1,1), (2,8), (3,27), \ldots\}$
4	$f_4(n) = \{(0,0), (1,5), (2,10), (3,15), \ldots\}$
5	$f_5(n) = \{(0,0), (1,10), (2,20), (3,30), \ldots\}$
….	…
	$f^1(n) = \{(0,1), (1,2), (2,5), (3,28), \ldots\}$

$f^1 \neq f_0 \quad f^1 \neq f_1 \quad f^1 \neq f_2 \quad f^1 \neq f_3 \ldots\ldots$

No :	Function, f(n) N→N
0	Your text here $f_0(n) = \{(0,0), (1,2), (2,4), (3,6), \ldots\}$
1	$f_1(n) = \{(0,0), (1,1), (2,2), (3,3), \ldots\}$
2	$f_2(n) = \{(0,0), (1,1), (2,4), (3,9), \ldots\}$
3	$f_3(n) = \{(0,0), (1,1), (2,8), (3,27), \ldots\}$
4	$f_4(n) = \{(0,0), (1,5), (2,10), (3,15), \ldots\}$
5	$f_5(n) = \{(0,0), (1,10), (2,20), (3,30), \ldots\}$
….	…
	$f^1(n) = \{(0,1), (1,2), (2,5), (3,28), \ldots\}$

$$f^1 \neq f_0 \quad f^1 \neq f_1 \quad f^1 \neq f_2 \quad f^1 \neq f_3 \ldots\ldots$$

- **Thus f^1 *does not align to any n Є N . Thus, Uncountable !!!***

No :	Function, f(n) N→N
0	Your text here $f_0(n) = \{(0,0), (1,2), (2,4), (3,6), \ldots\}$
1	$f_1(n) = \{(0,0), (1,1), (2,2), (3,3), \ldots\}$
2	$f_2(n) = \{(0,0), (1,1), (2,4), (3,9), \ldots\}$
3	$f_3(n) = \{(0,0), (1,1), (2,8), (3,27), \ldots\}$
4	$f_4(n) = \{(0,0), (1,5), (2,10), (3,15), \ldots\}$
5	$f_5(n) = \{(0,0), (1,10), (2,20), (3,30), \ldots\}$
….	…
	$f^1(n) = \{(0,1), (1,2), (2,5), (3,28), \ldots\}$