CSE 2213: Discrete Mathematics

Section - O Room No - 325

Prof. Dr. A.K.M. Muzahidul Islam

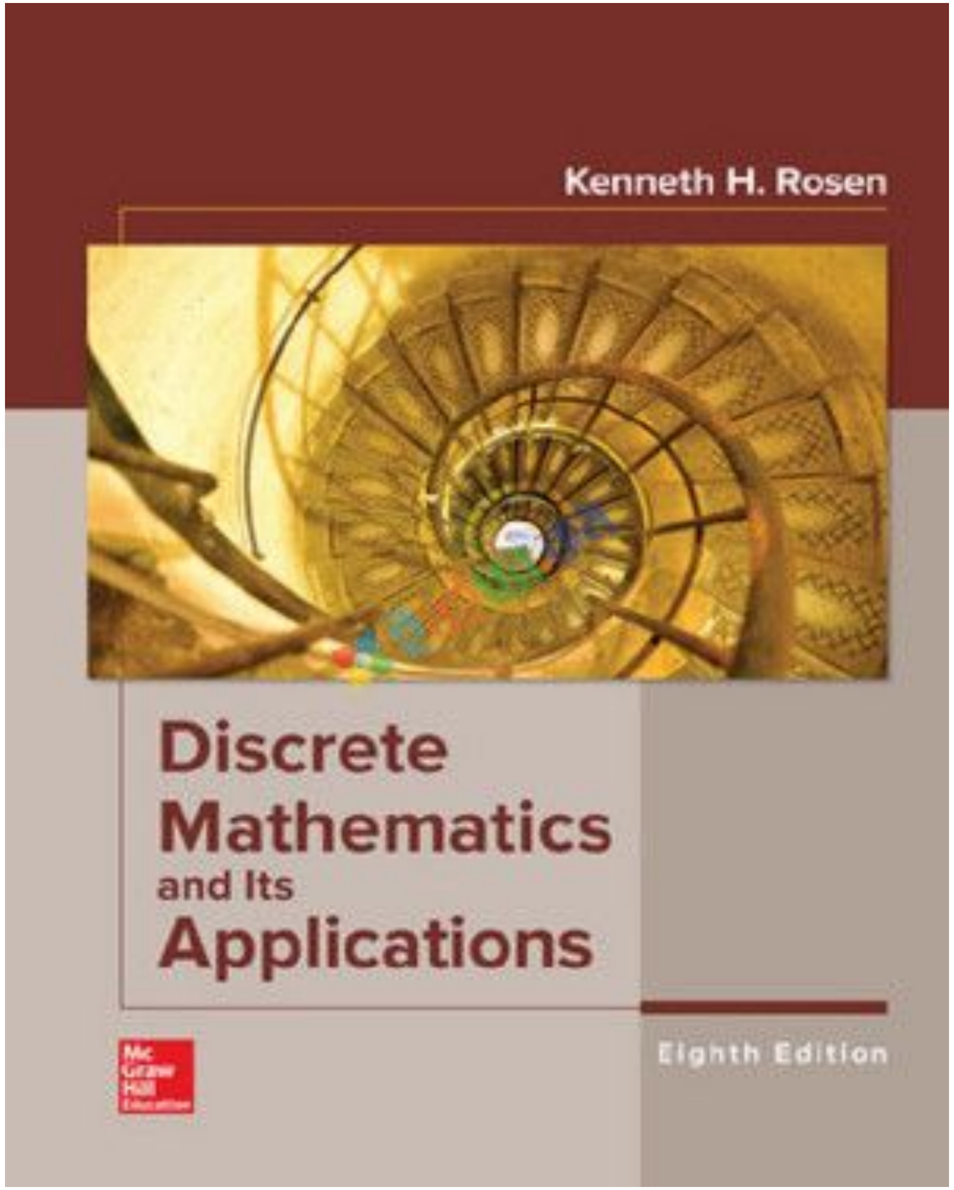
Computer Science & Engineering (CSE)

United International University (UIU)

Email: muzahid@cse.uiu.ac.bd

Fall 2024

Main Textbook



Discrete Mathematics and Its Applications: 8th edition
Kenneth H. Rosen

Course Outline: Summary

1. Chapter 1: The Foundations
   Propositional Logic, Predicates, Quantifiers and Proofs
2. Chapter 2: Basic Structures:
   Sets, Functions, Sequences, Sums, etc.
3. Chapter 5: Induction
4. Chapter 6: Counting Basics and The Pigeonhole Principle
5. Chapter 10: Graph
6. Chapter 11: Tree

Assessment

- Attendance : 5% (5: 22>= ; 4: 19>= ; 3: 16>=)
- Class Tests : 20% (Best 03 Out of 04 Class Test)
    - Written Class Test
- Assignment : 5%
- Mid Term Exam : 30%
- Final Assessment : 40%

Grading Policy

A (Plain)	: 90-100	C+ (Plus)	: 70-73
A- (Minus)	: 86-89	C (Plain)	: 66-69
B+ (Plus)	: 82-85	C- (Minus)	: 62-65
B (Plain)	: 78-81	D+ (Plus)	: 58-61
B- (Minus)	:74-77	D (Plain)	: 55-57

Outline

- Introduction to Discrete Mathematics
- Some Examples: Example 1, 2 and 3
- Propositional Logic
- Truth Table

Discrete Mathematics

Introduction to Discrete Mathematics

- **What is Discrete Mathematics?**
  - Is the Study of Discrete Objects i.e. those are not Continuous Objects.
  - **Discrete:**
    - Cannot be broken down into fractions or decimals.
    - Discrete elements are countable.
    - **Example:** Number of Students, Tickets, etc.
  - **Continuous:**
    - Can be broken down into Fractions or Decimals.
    - Continuous elements are measurable.
    - Example: Time, Temperature, Distance, etc.
- **Why do you need to study DM?**
  - It is a very good tool for improving problem-solving capabilities.
    - Logical reasoning
    - Mathematical proofs
    - Complexity of algorithms

Applications

- Data Structure
- Algorithms
- Operating Systems
- Databases
- Software Engineering
- Operating Systems
- Computer Networks

PROPOSITIONAL LOGIC

- **The rules of logic specify the meaning of mathematical statements.**
- For instance, these rules help us understand and reason with statements
- Logic is the basis of all mathematical reasoning.
- It has practical applications to the design of computing machines to computer programming languages, as well as to many other fields of study.
- Example 1 : "There exists an integer that is not the sum of two squares".
- Example 2 : "For every positive integer ***n***, the sum of the positive integers not exceeding ***n*** is $n(n + 1)/2$".
- Example 3 : **"An undirected graph has an Eulerian cycle if and only if**
  (1) Every node degree is even and
  (2) the graph is connected (a path exists between nodes)".

Example No. 1

“There exists an integer that is not the sum of two squares”

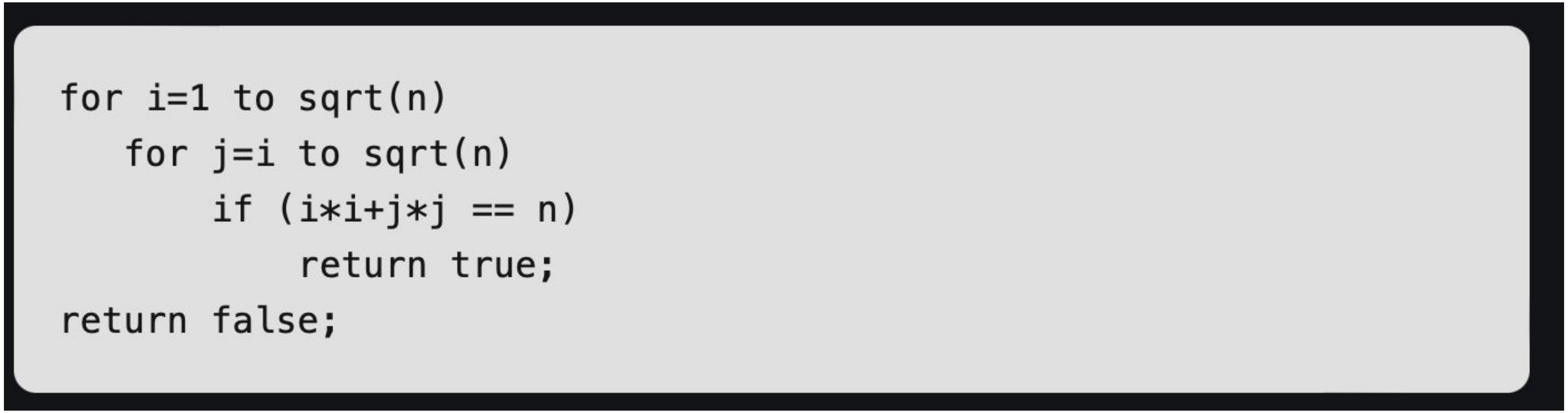
- Let’s consider that we have integer number ***n*** *e.g.* ***16*****,** ***18*****,** ***24*****,** ***169***.
- We need to find **whether the number *n* can be represented by the sum of two squares**.

  - For example n = 16
    - **Input n : 16**
    - **Output : No**
  - For example n = 24
    - **Input n : 24**
    - **Output : No**
  - **For example n = 18,**
    - **Input n : 18**
    - **Output : Yes**
    - 3 ^ 2 + 3 ^ 2 = 9 + 9 = 18
  - For example n = 169
    - **Input n : 169**
    - **Output : Yes**
    - 5 ^ 2 + 12 ^ 2 = 25 + 144 = 169

Example No. 1 (Cont'd)

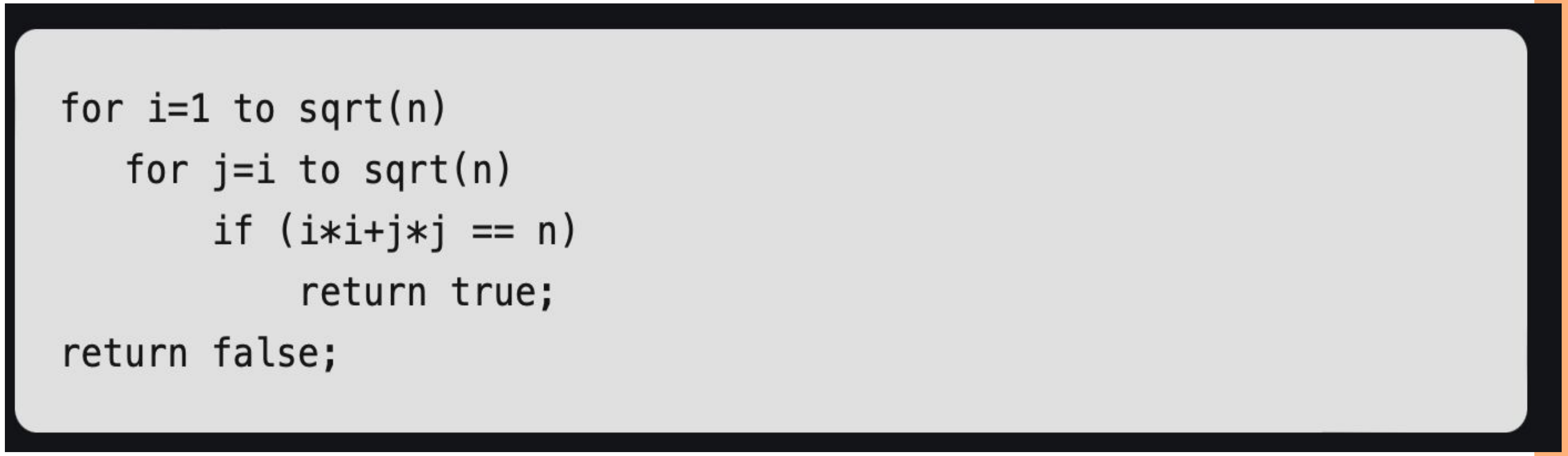
“There exists an integer that is not the sum of two squares”

- **We use 2 Loops running till the Square Root of *n*.**
- Each time, we find whether the Sum of the Square of Both Numbers of the Loop is Equal to ***n.***
- **If we find that Combination, we Print Yes, other we Print No**



```
for i=1 to sqrt(n)
   for j=i to sqrt(n)
       if (i*i+j*j == n)
           return true;
return false;
```

Example No. 1 (Cont'd) Case n = 18



```
for i=1 to sqrt(n)
   for j=i to sqrt(n)
       if (i*i+j*j == n)
           return true;
return false;
```

1. i = 1

    j = 1, 1*1 + 1*1 = 2
    j = 2, 1*1 + 2*2 = 5
    j = 3, 1*1 + 3*3 = 10
    j = 4, 1*1 + 4*4 = 17
    NOT FOUND YET

1. i = 2

    j = 2, 2*2 + 2*2 = 8
    j = 3, 2*2 + 3*3 = 13
    j = 4, 2*2 + 4*4 = 20
    NOT FOUND YET

1. i = 3

    j = 3, 3*3 + 3*3 = 18 YES
    ALREADY FOUND

Example No. 2

"For every positive integer *n*, the sum of the positive integers not exceeding *n* is *n*(*n* + 1)/2".

- Let's Prove by **Mathematical Induction** that

$$1 + 2 + 3 + \ldots + n = n * (n + 1)/2.$$

- **Base Case:** $n = 1$ L.H.S R.H.S

$$= 1 \qquad = 1 * (1 + 1) / 2$$

$$= 1 * (2) / 2 = 1$$

- **Induction Case:** Assume that the above statement is TRUE for n = k
  - $1 + 2 + 3 + \ldots + k = k * (k + 1)/2.$
  - **Prove that it is True for n = k + 1**

$$1 + 2 + 3 + \ldots k + k{+}1 = (k + 1) * ((k + 1) + 1)/2.$$

$$1 + 2 + 3 + \ldots k + (k{+}1) = (k + 1) * (k + 2)/2$$

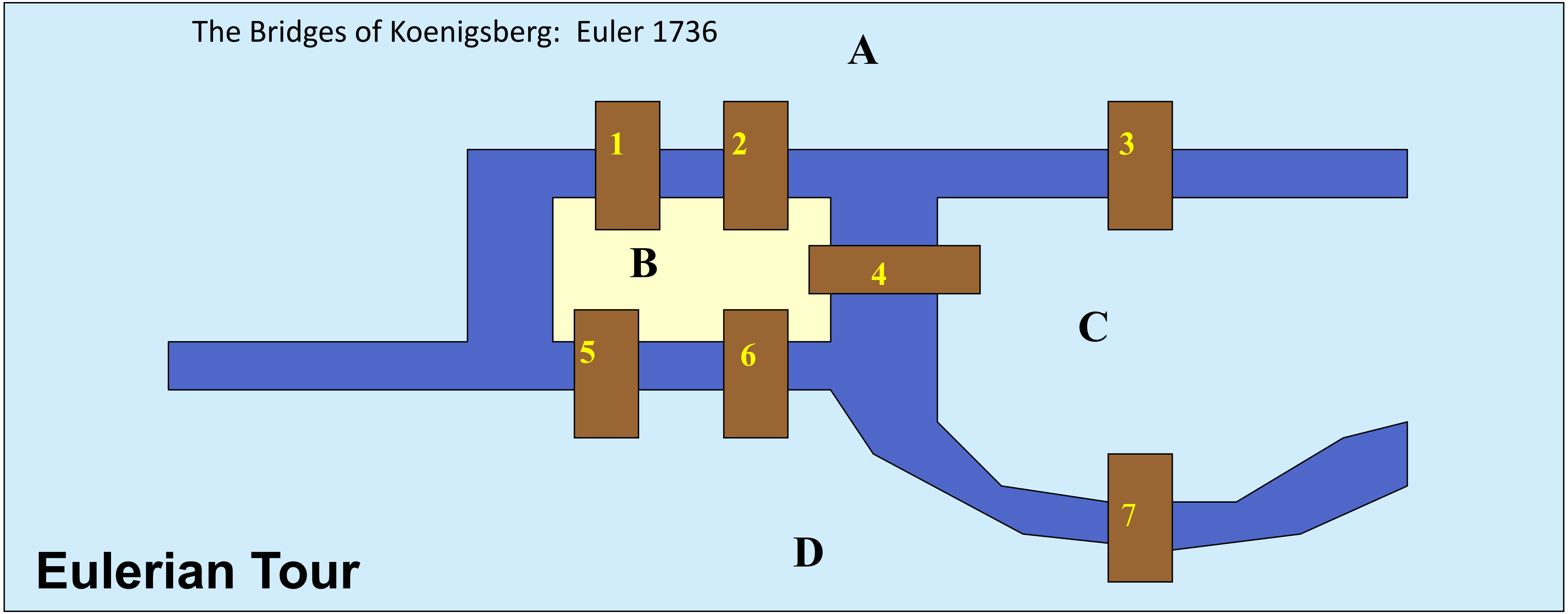
$$k * (k + 1)/2 + (k{+}1) = (k \wedge 2 + 2k + k + 2)/2$$

$$(k \wedge 2 + k + 2k{+}2\ ) / 2 = (k \wedge 2 + k + 2k + 2)/2$$

Example No. 3

“An undirected graph has an eulerian cycle if and only if
(1) Every node degree is even and
(2) the graph is connected (a path exists between nodes)”.

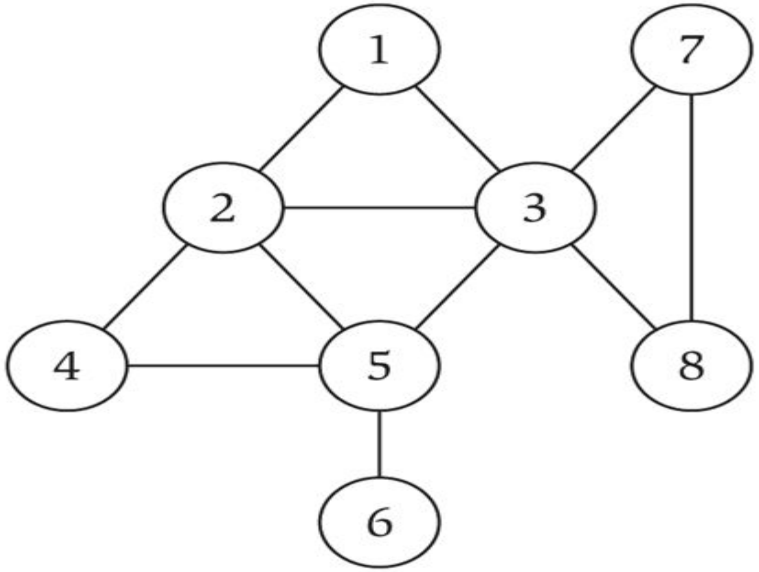
The Bridges of Koenigsberg: Euler 1736



Eulerian Tour

Is it possible to start in A, cross over each bridge exactly once, and end up back in A?

- **Graph** – mathematical object consisting of a set of:
  - *V* = **nodes** (vertices, points).
  - *E* = **edges** (links, arcs) between pairs of nodes.
  - Denoted by *G = (V, E).*
  - **Graph size** parameters: *n* = |*V*|, *m* = |*E*|.

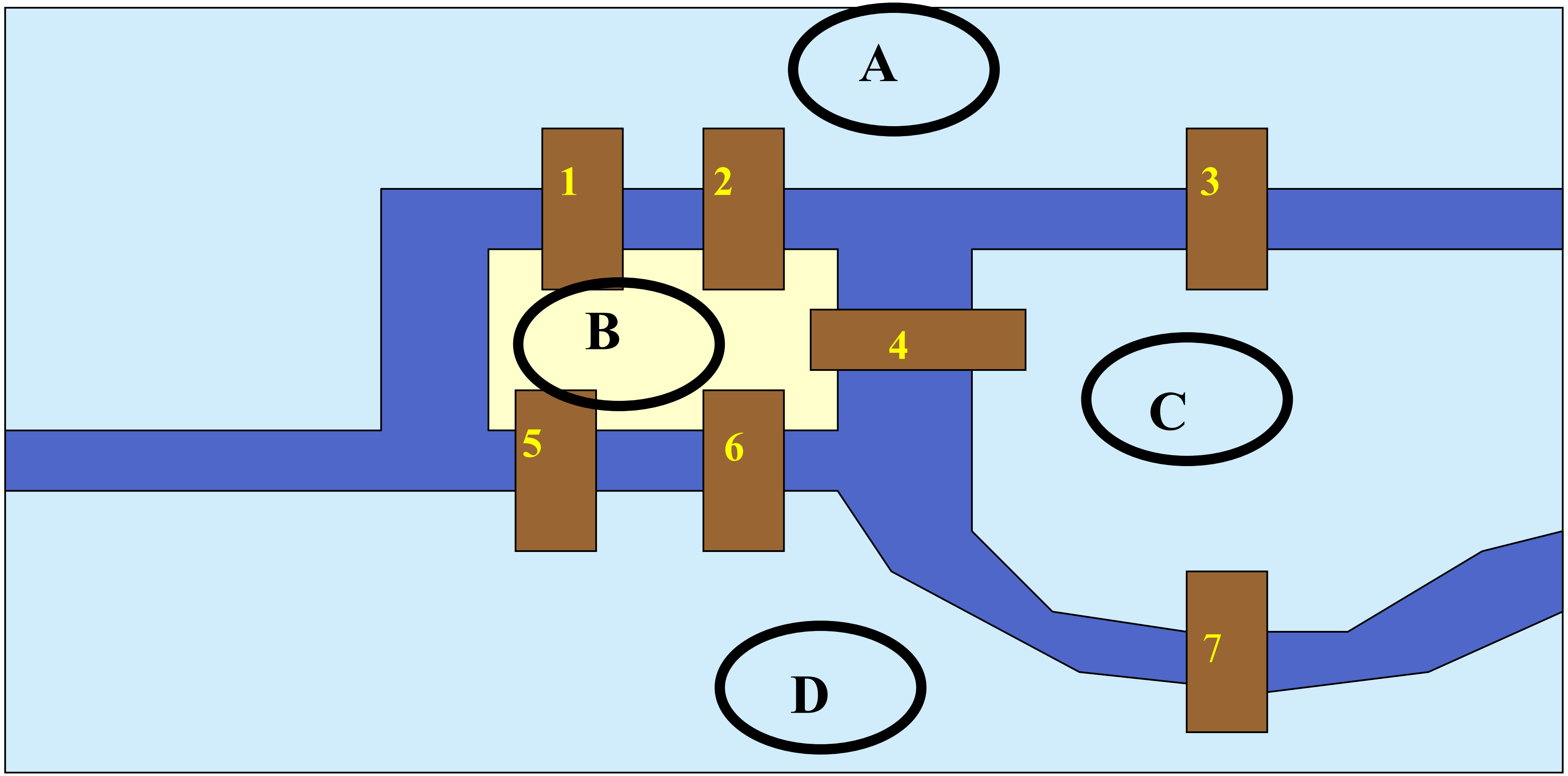


V = { 1, 2, 3, 4, 5, 6, 7, 8 }

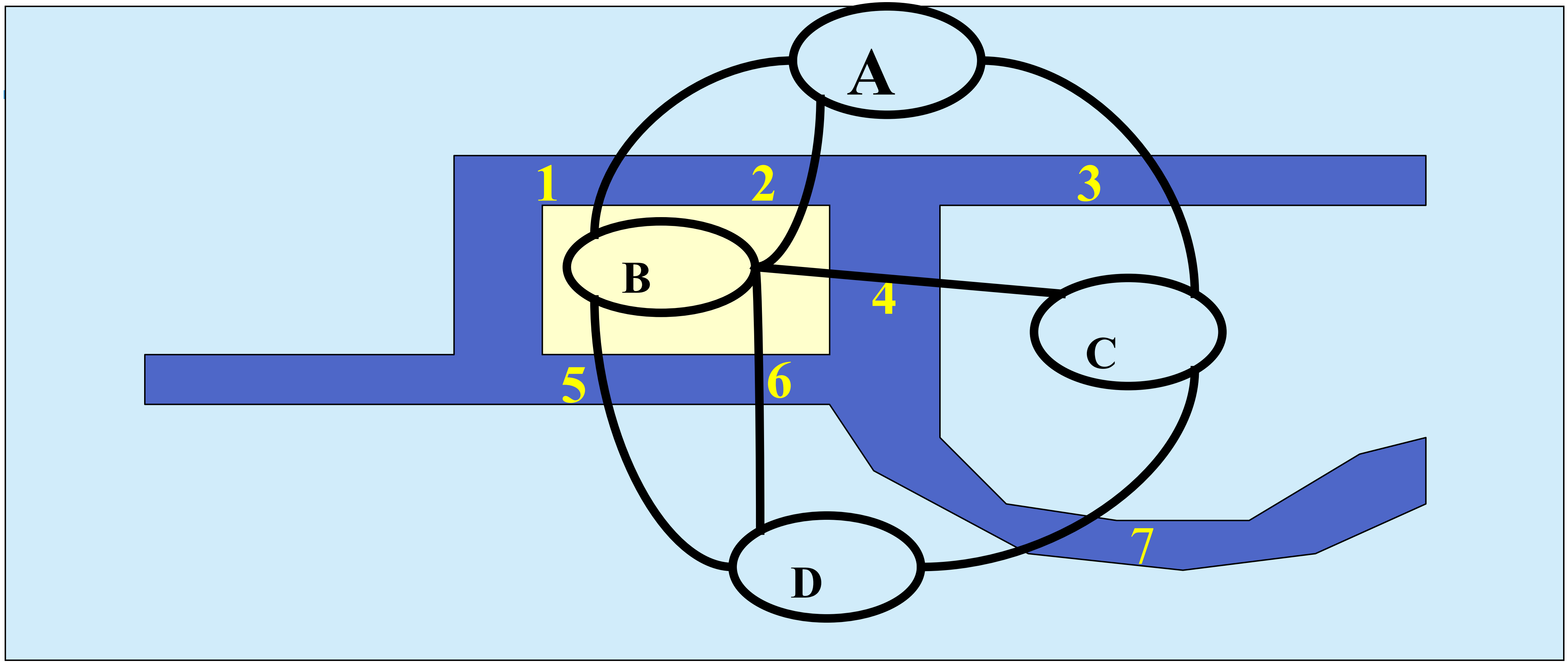
E = { {1,2}, {1,3}, {2,3}, {2,4}, {2,5}, {3,5}, {3,7}, {3,8}, {4,5}, {5,6} }

n = 8

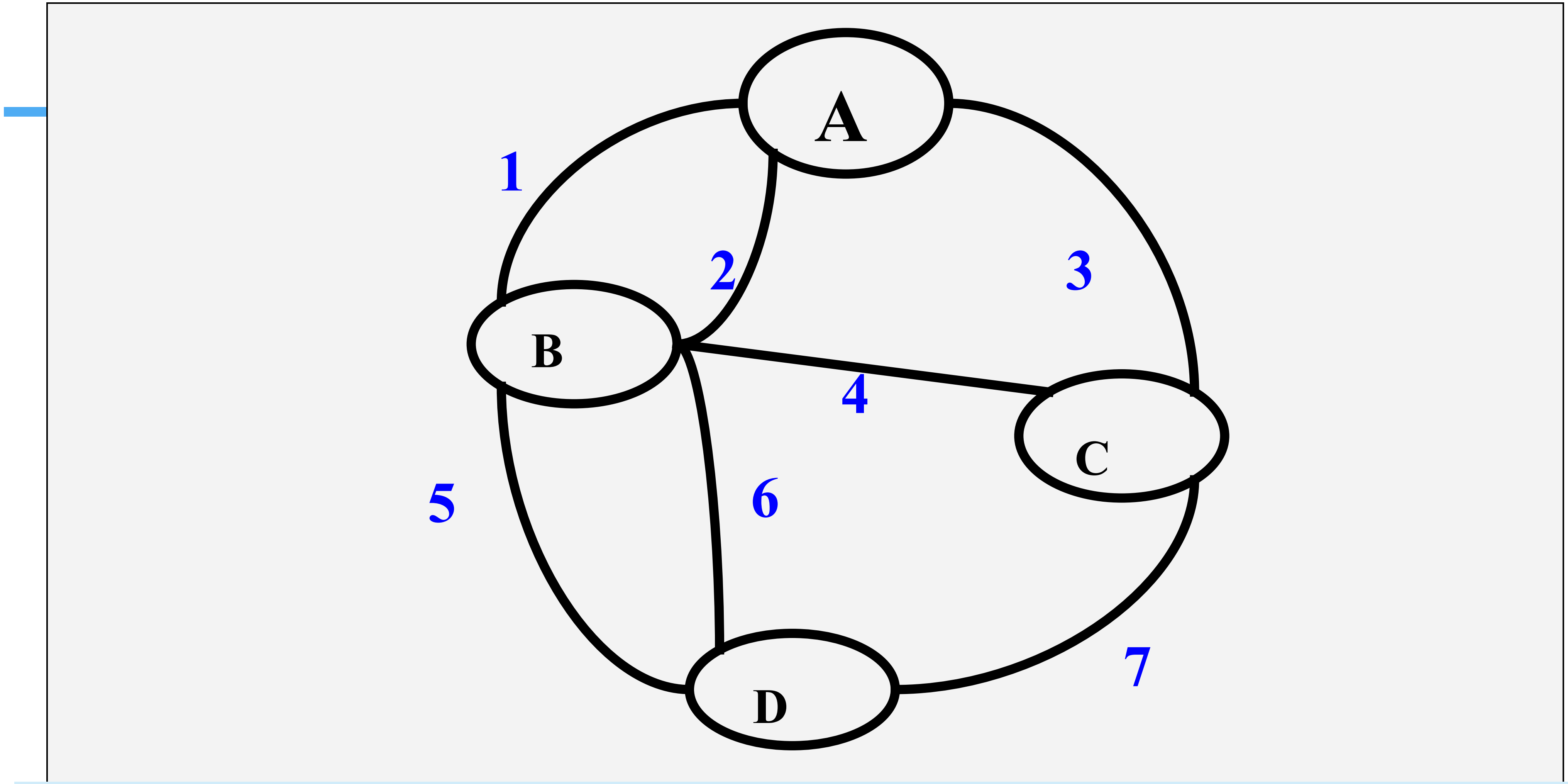
m = 11



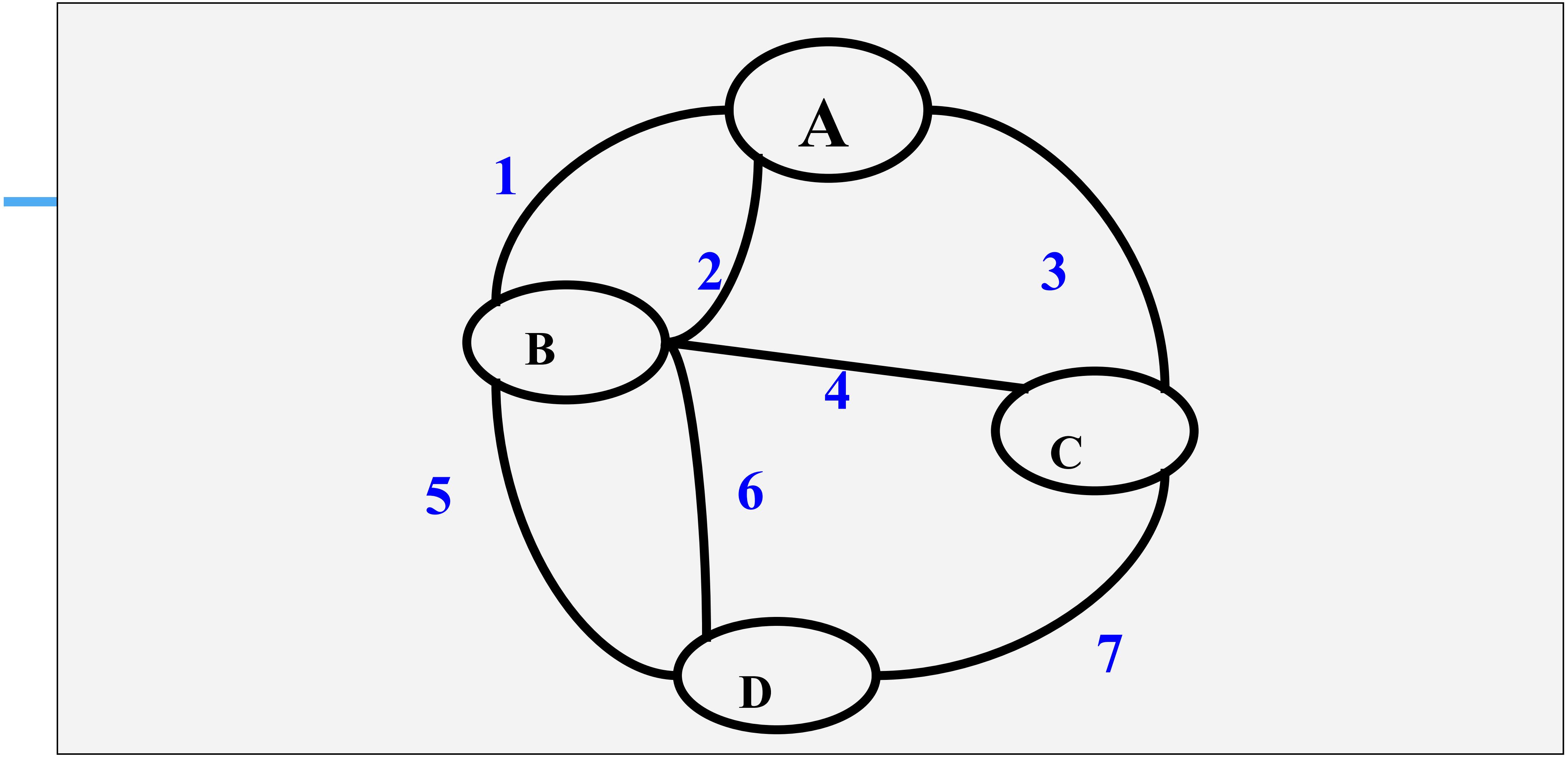
Conceptualization: Land masses are “nodes”.



Conceptualization: Bridges are “arcs.”

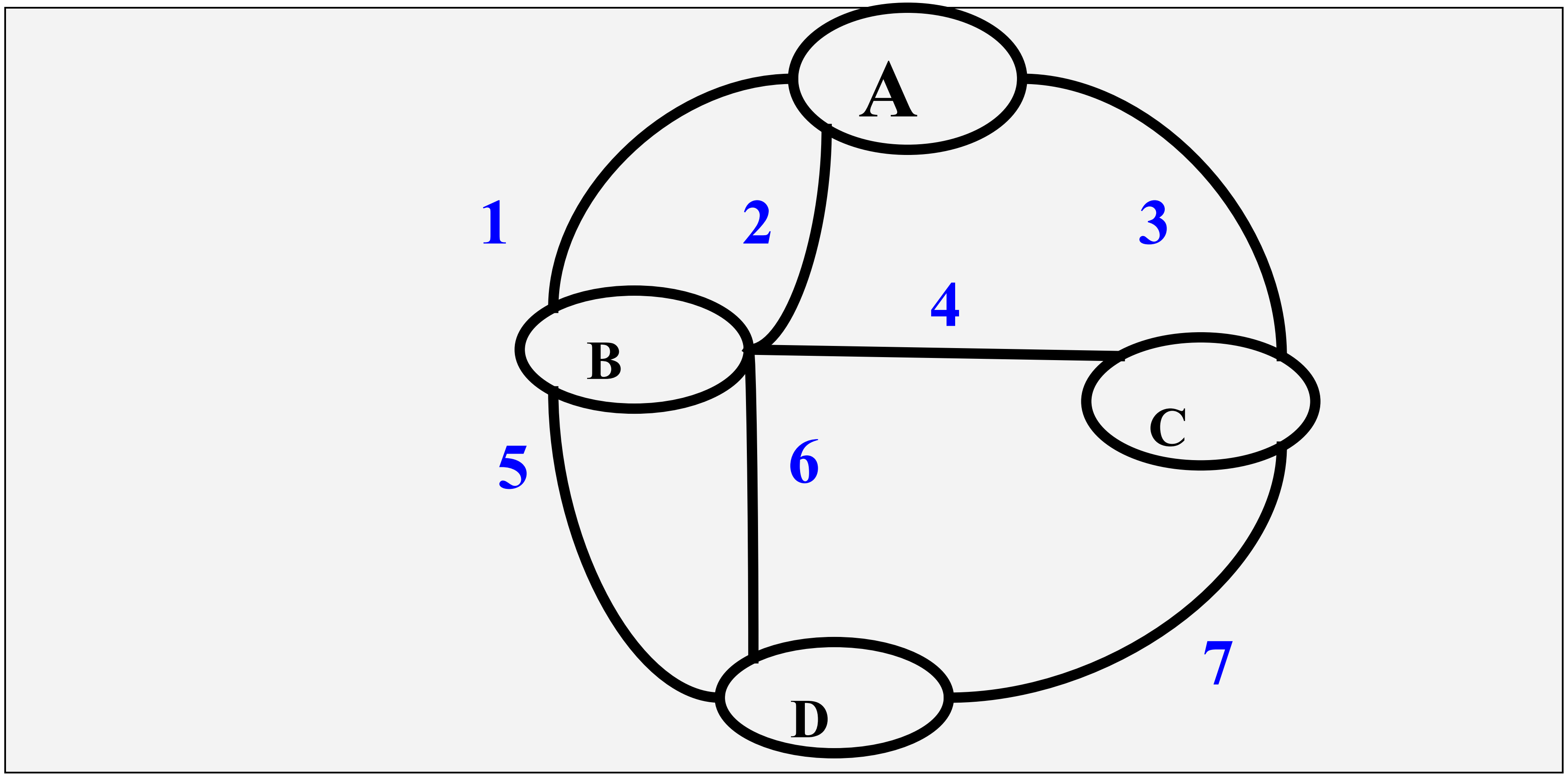


Is there a “walk” starting at A and ending at A and passing through each arc exactly once?

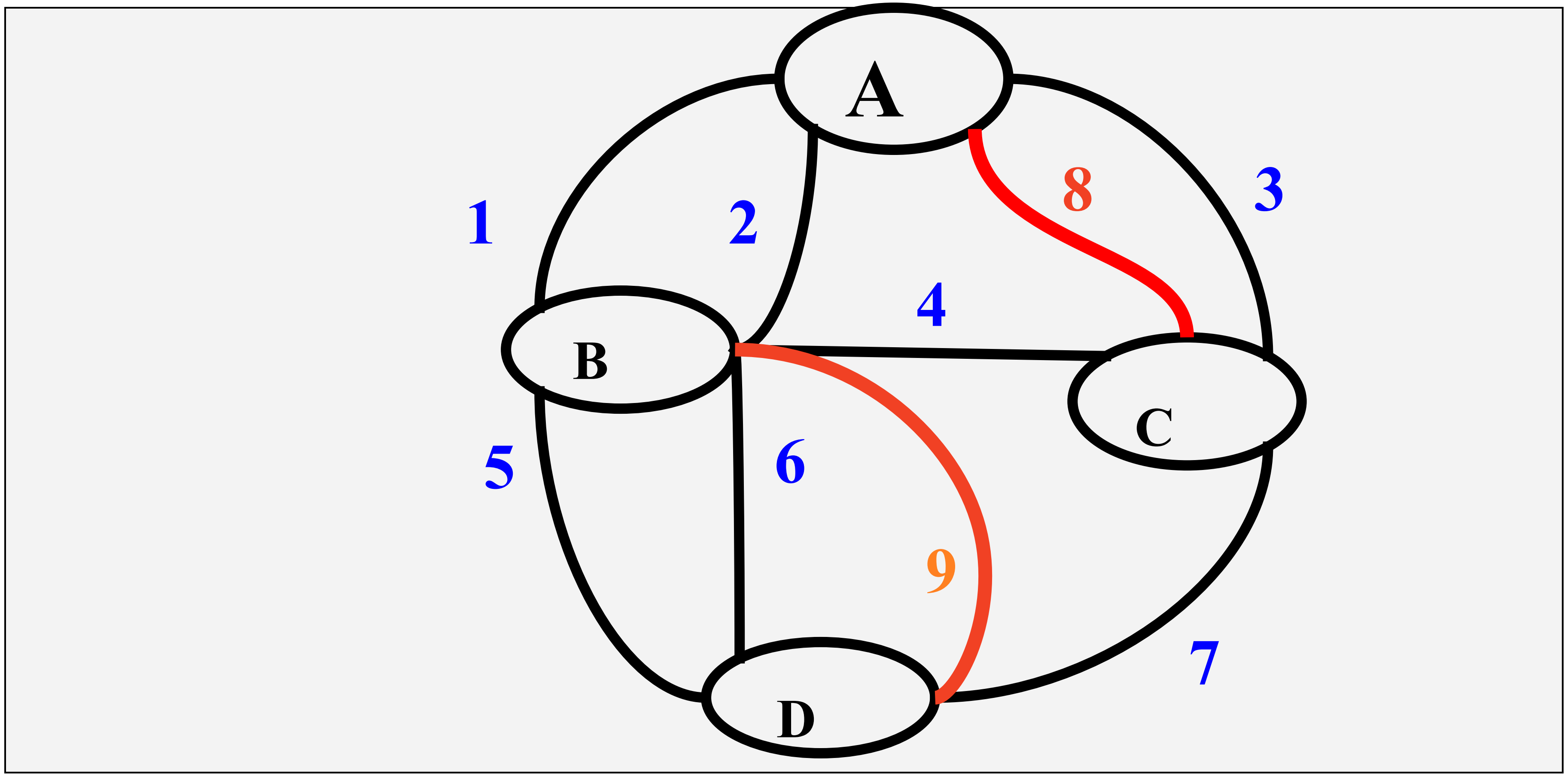


Is there a "walk" starting at A and ending at A and passing through each arc exactly once?

Such a walk is called an *eulerian cycle*.



Is there a “walk” starting at A and ending at A and passing through each arc exactly once?
Such a walk is called an *eulerian cycle*.



Adding two bridges creates such a walk

Walk: A, 1, B, 5, D, 6, B, 4, C, 8, A, 3, C, 7, D, 9, B, 2, A

Note: the number of arcs incident to B is twice the number of times that B appears on the walk.

Example No. 3 (Cont'd)

The degree of a node in an undirected graph is the number of incident arcs

Theorem. An undirected graph has an eulerian cycle if and only if

(1) every node degree is even and

(2) the graph is connected (that is, there is a path
from each node to each other node).

PROPOSITION

A proposition is a declarative sentence

- It can be either true or false
- It cannot be neither
- It cannot be both

Let p be "Washington DC is the capital of USA",
and q be "Toronto is the capital of Canada"

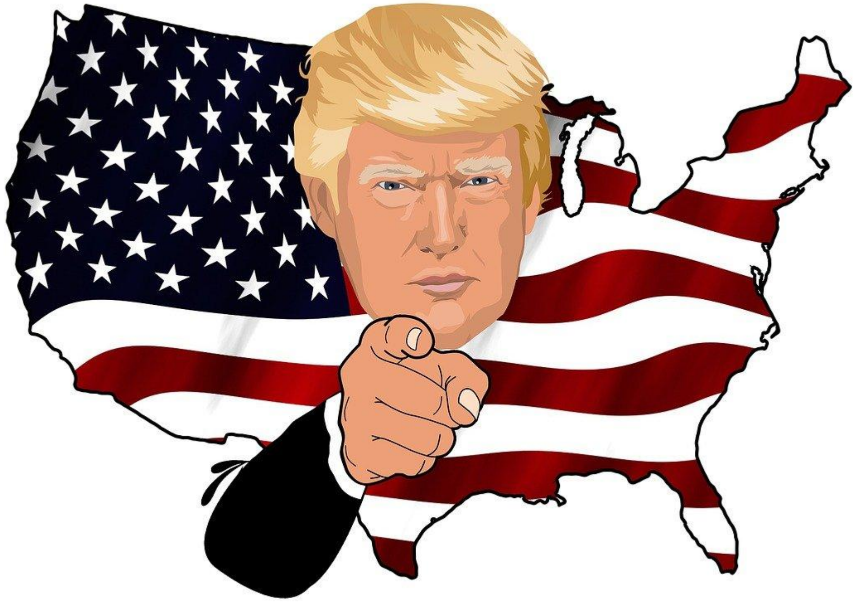
- p is a true proposition
- q is a false proposition

A proposition can have one of two truth values – **true** or **false**

Propositional Variables

- We **use letters** to denote **propositional variables** (or **sentential variables**).
- The conventional letters used for propositional variables are ***p, q, r, s, ...***
- The **truth value** of a proposition
  - **is true, denoted by T**, if it is a true proposition, and
  - **is false, denoted by F,** if it is a false proposition.
- Propositions that cannot be expressed in terms of simpler propositions are called **atomic propositions**.
- **The area of logic** that deals with propositions is called the **propositional calculus** or **propositional logic**.
- Developed by the **Greek philosopher Aristotle more than 2300 years ago.**

WHICH ONE IS A PROPOSITION?



Let *p* be “I’m Bill Clinton”

- Is it a Proposition?
- p is True / False Proposition?

Let *q* be “I’m Hilary Clinton”

- Is it a Proposition?
- q is True / False Proposition?

WHAT IS NOT A PROPOSITION?

What is your name?

- A question – hence not a declarative sentence

Tell me your name.

- An order – hence not a declarative sentence

$x + 1 = y$

- An equation – hence neither true nor false
- Potential proposition with gap(s) → Predicate

$x + 1 = y$, given that $x = 5$ and $y = 6$

- An equation with values of each variable – a proposition
- If you fill up all the gaps of a predicate, you get a proposition

WHAT IS NOT A PROPOSITION?

I am lying.

- If this sentence is true, then since I am saying this, it must be false!!!
- **Both true and false**
  - **Not a proposition**
- We call such a sentence a paradox

COMPOUND PROPOSITION

Taskin ahmed is NOT a batsman

Sachin Tendulkar will bat AND Mustafiz will bowl

Shariful Islam will bowl OR Nasum Ahmed will bowl

IF top order performs, THEN the team will win

Bangladesh will **win** IF AND ONLY IF the players are well prepared

COMPOUND PROPOSITION

Two (or more) propositions combined or modified with logical connectives

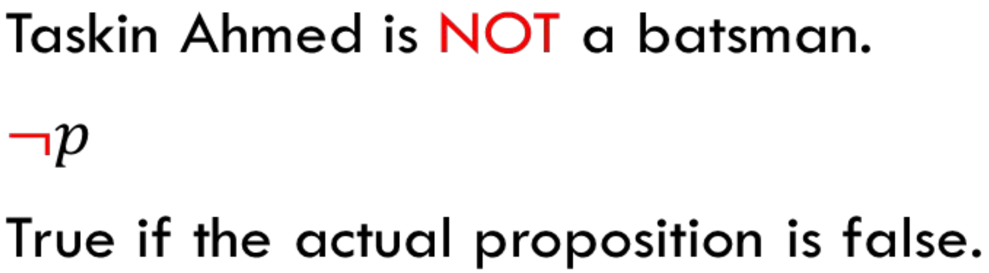
- NOT (¬) – Negation
- AND (□) – Conjunction
- OR (□) – Disjunction
- If … then (→) – Conditional statement
- If and only if (*iff*) – Biconditional statement

NEGATION

Definition 1: Let *p* be a proposition.
The *negation of p*, denoted by ¬*p*
(also denoted by ~p), is the statement
"It is not the case that *p*."

The proposition ¬*p* is read "not *p*."

The truth value of the negation of *p*,
¬*p*, is the opposite of the truth value of
p.



Taskin Ahmed is NOT a batsman.

$\neg p$

True if the actual proposition is false.

p	¬*p*
T	F
F	T

CONJUNCTION

Definition 2: Let *p* and *q* be propositions.

The *conjunction* of *p* and *q*, denoted by *p* ∧ *q*, is the proposition "*p* and *q*."

The conjunction *p* ∧ *q* is

- **True, when both *p* and *q* true**
- **False**, otherwise.

Virat Kohli will bat AND Mustafiz will bowl.

$p \wedge q$

True if both propositions are true

Truth table of $p \wedge q$

p	*q*	*p* ∧ *q*
T	T	T
T	F	F
F	T	F
F	F	F

DISJUNCTION

Definition 3: Let *p* and *q* be propositions.

The *disjunction* of *p* and *q*, denoted by *p* ∨ *q*, is the proposition "*p* or *q*."

The disjunction *p* ∨ *q* is

- **False, if both *p* and *q* false**
- **True, otherwise.**

Shariful Islam will bowl OR Saifuddin will bowl.

$p \vee q$

True if any proposition is true

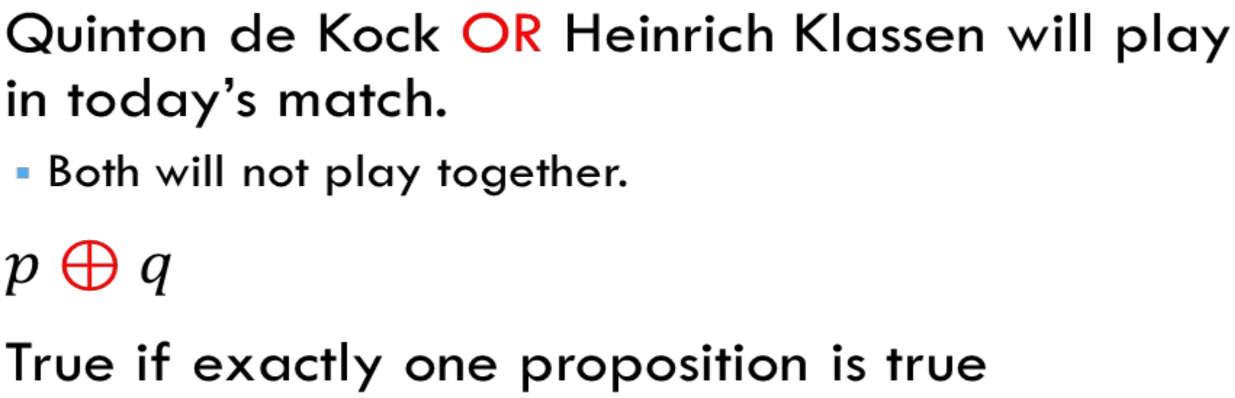
p	*q*	*p* ∨ *q*
T	T	T
T	F	T
F	T	T
F	F	F

A SPECIAL TYPE OF DISJUNCTION – EXCLUSIVE-OR

Definition 4: **Let *p* and *q* be propositions.**

The *exclusive or* of *p* and *q*, denoted by $p \oplus q$ (or *p* XOR *q*), is the proposition that is

- **True when exactly one of *p* and *q* is true and**
- **False, otherwise.**



Quinton de Kock OR Heinrich Klassen will play in today's match.

- Both will not play together.

$p \oplus q$

True if exactly one proposition is true

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

TRUTH TABLES OF COMPOUND PROPOSITIONS (Exercise No. 1)

A truth table is used to observe the behavior of a compound proposition at a glance

The different possible truth values of the simple propositions are compiled in the table, and for each combination, the truth value of the compound one is found out

p	*q*	Truth Value	□	□	⊕
T	T	T	?	?	?
T	T	F	?	?	?
T	F	T	?	?	?
T	F	F	?	?	?
F	T	T	?	?	?
F	T	F	?	?	?
F	F	T	?	?	?
F	F	F	?	?	?

Exercise No. 2

Construct the truth table of the proposition $(p \lor \neg q) \land r$.

CONDITIONAL PROPOSITIONS

Definition 5: Let *p* and *q* be propositions.

The *conditional statement p* → *q* is the proposition "if *p*, then *q*."

The conditional statement *p* → *q* is

- **False when *p* is true and *q* is false, and**
- **True, otherwise.**

In the conditional statement *p* → *q*,

- ***p* is called the *hypothesis*** (or *antecedent* or *premise*) and
- ***q* is called the *conclusion* (or *consequence*)**.

Conditional Statement in Programming Language

- **Programming Language:**
    - if (p) then
        - { S; }

  p is a proposition and S is a program segment

  S is not a proposition

- **Example:** What is the value of the variable x after the statement
    - **if** 2 + 2 = 4 **then** $x := x + 1$
    - if $x = 0$ before this statement is encountered?
        - The symbol := stands for assignment.
        - $x := x + 1$ means the assignment of the value of $x + 1$ to x.
- **Solution: Because 2 + 2 = 4 is true,**
    - **the assignment statement $x := x + 1$ is executed.**
    - Hence, x has the value 0 + 1 = 1 after this statement is encountered.

IF IT RAINS, I WILL STAY HOME

The sentence is true if it is raining, and you are staying home.

WHAT IF IT DOESN’T RAIN?

6/22/2024

IF IT RAINS, I WILL STAY HOME

If it does not rain, you might still stay home
(probably because you are sick)

IF IT RAINS, I WILL STAY HOME

The sentence does not say anything about what if it does not rain

So if it does not rain, we can say you have not broken any condition

So if it does not rain, the sentence is true whether you stay home or not

SO WHEN IS THIS SENTENCE FALSE?

IF IT RAINS, I WILL STAY HOME

The sentence is false if it rains, and you go out anyway

CONDITIONAL STATEMENTS

If it rains, I will stay home.

- We can also say “It is raining implies that I will stay home”

$p \rightarrow q$

- If p then q
- p implies q
- q if p

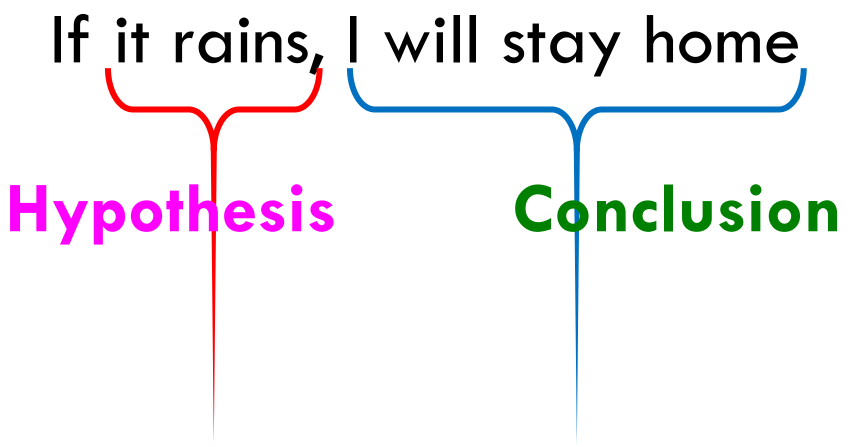
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

HYPOTHESIS AND CONCLUSION

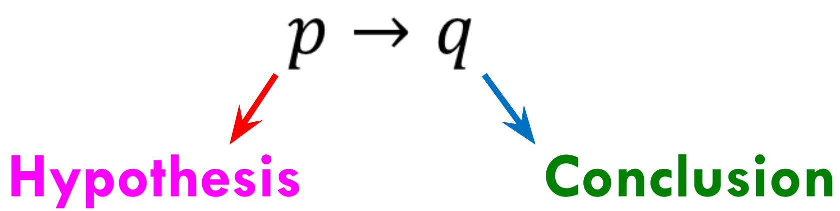
The if clause is called the hypothesis, and the remaining clause is called the conclusion

We assume that the hypothesis is true in order to verify the validity of the conclusion

The conclusion is the outcome of the hypothesis



If it rains, I will stay home



$p \to q$

BICONDITIONAL STATEMENT

Expresses that 2 propositions have the same truth value.

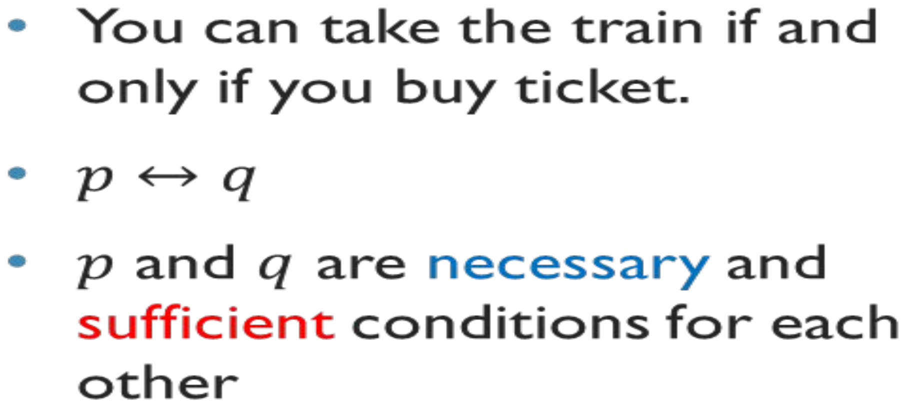
Definition 6: Let *p* and *q* be propositions.

The *biconditional statement p* ↔ ***q*** is the proposition "***p* if and only if *q*.**"

The biconditional statement *p* ↔ ***q* is**

- **True** when *p* and *q* have the same truth values, and
- **False**, otherwise.

Biconditional statements are also called *bi-implications*.



- You can take the train if and only if you buy ticket.
- $p \leftrightarrow q$
- p and q are necessary and sufficient conditions for each other

p	*q*	*p* ↔ *q*
T	T	T
T	F	F
F	T	F
F	F	T

CONVERSE, CONTRAPOSITIVE AND INVERSE

- We can form some new conditional statements starting with a conditional statement $p \rightarrow q$.
- **There are 3 related conditional statements** that occur so often that they have special names.

$p \rightarrow q$	If it rains, I will stay home.
$q \rightarrow p$	**Converse** : If I stay home, it is raining.
$\neg q \rightarrow \neg p$	**Contrapositive:** If I do not stay home, it is not raining.
$\neg p \rightarrow \neg q$	**Inverse** : If it does not rain, I will not stay home.

Which of these are equivalent?

PRECEDENCE OF OPERATORS

Rule 1: Generally we use parentheses to specify the order in which logical operators in a compound proposition are to be applied.

For instance, $(p \lor q) \land (\neg r)$ is the conjunction of $p \lor q$ and $\neg r$.

To reduce the number of parentheses, we specify that the negation operator is applied before all other logical operators.

This means that $\neg p \land q$ is the conjunction of $\neg p$ and q, namely, $(\neg p) \land q$, not the negation of the conjunction of p and q, namely $\neg(p \land q)$.

- **i.e., $(\neg p) \land q \neq \neg(p \land q)$**

Rule 2: Another general rule of precedence is that **the conjunction operator takes precedence over the disjunction operator,**

so, $p \lor q \land r$ means

$p \lor (q \land r)$ rather than $(p \lor q) \land r$

Ex.1: $p = T$, $q=T$ and $r=F$ SOLVE

and $p \land q \lor r$ means

$(p \land q) \lor r$ rather than $p \land (q \lor r)$.

Ex.2: $p = F$, $q=T$ and $r=T$ SOLVE

PRECEDENCE OF OPERATORS (CONT'D)

Operator	Precedence
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

Rule 3: Conditional and biconditional operators, $\rightarrow$ and $\leftrightarrow$, have

lower precedence than the conjunction and disjunction operators, $\wedge$ and $\vee$.

Consequently,

$p \rightarrow q \vee r$ means $p \rightarrow (q \vee r)$

rather than $(p \rightarrow q) \vee r$ and

$p \vee q \rightarrow r$ means $(p \vee q) \rightarrow r$

rather than $p \vee (q \rightarrow r)$.

Logic and Bit Operations

- Computers represent information using bits.
- **A bit is a symbol with two possible values, namely,**
  - **0 (zero) and 1 (one).**
- **A bit can be used to represent a truth value, namely,**
  - **true and false.**
  - **1 represents T (true), and 0 represents F (false).**
- **A variable is called a Boolean variable**
  - if its value is either true or false.
  - A Boolean variable can be represented using a bit.

- **By replacing true by a one and false by a zero in the truth tables for the operators ∧, ∨, and ⊕,** the columns in Table 9 for the corresponding bit operations are obtained.

- **The notation *OR*, *AND*, and *XOR* for the operators ∨, ∧, and ⊕, are used as in various programming languages.**

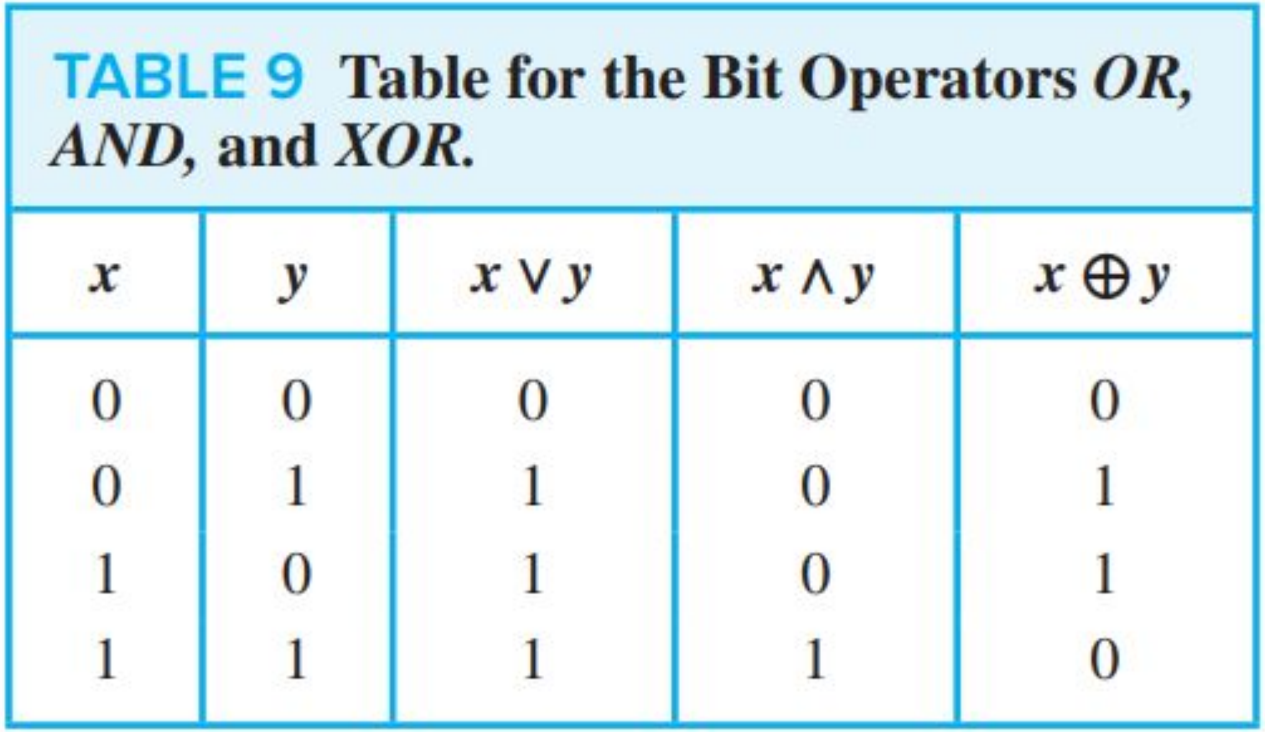


TABLE 9 **Table for the Bit Operators *OR*, *AND*, and *XOR*.**

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Example

Find the bitwise OR, bitwise AND, and bitwise XOR of the bit strings 01 1011 0110 and 11 0001 1101. (Here, and throughout this book, bit strings will be split into blocks of four bits to make them easier to read.)

Solution: The bitwise OR, bitwise AND, and bitwise XOR of these strings are obtained by taking the OR, AND, and XOR of the corresponding bits, respectively.

This gives us

01 1011 0110

11 0001 1101

---

11 1011 1111 bitwise OR

01 0001 0100 bitwise AND

10 1010 1011 bitwise XOR

1.3 Propositional Equivalences

- An important type of step used in a mathematical argument is the **replacement of a statement with another statement with same truth value.**

- **Definition 1: A compound proposition that is always true,**

  no matter what the truth values of the propositional variables that occur in it, is called a *tautology*.

  A compound proposition that is always false is called a *contradiction*.

  A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

Example

We can construct examples of tautologies and contradictions using just one propositional variable.

Consider the truth tables of $p \vee \neg p$ and $p \wedge \neg p$, shown in Table 1.

Because $p \vee \neg p$ is always true, it is a tautology.

Because $p \wedge \neg p$ is always false, it is a contradiction.

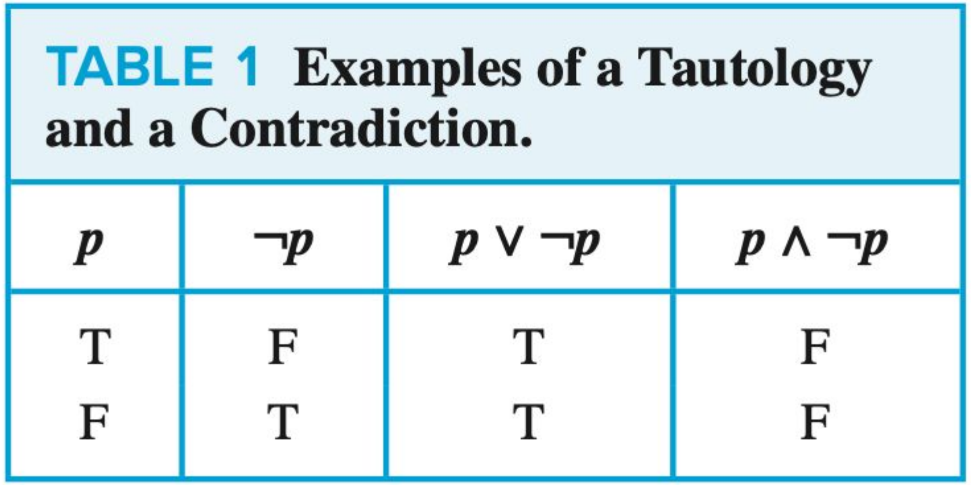


TABLE 1 Examples of a Tautology and a Contradiction.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

LOGICAL EQUIVALENCE

Compound propositions that have the same truth values in all possible cases are called **logically equivalent**.

Definition 2: The compound propositions *p* and *q* are called *logically equivalent* if $p \leftrightarrow q$ is a tautology.

The notation $p \equiv q$ denotes that *p* and *q* are logically equivalent.

Remark: The symbol ≡ is not a logical connective, and $p \equiv q$ is not a compound proposition but rather is the statement that $p \leftrightarrow q$ is a tautology.

The symbol ⇔ is sometimes used instead of ≡ to denote logical equivalence.

De Morgan laws

One way to determine **whether two compound propositions are equivalent** is to **use a truth table**.

The compound propositions p and q are equivalent if and only if the columns giving their truth values agree.

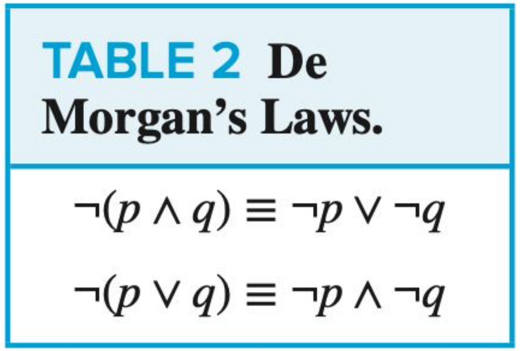


TABLE 2 De Morgan's Laws.
$\neg(p \land q) \equiv \neg p \lor \neg q$
$\neg(p \lor q) \equiv \neg p \land \neg q$

Example 2 illustrates this method to establish an extremely important and useful logical equivalence, namely, that of $\neg(p \lor q)$ with $\neg p \land \neg q$.

This logical equivalence is one of the two De Morgan laws, shown in Table 2, named after the English mathematician Augustus De Morgan, of the mid-nineteenth century.

Exercise

Example 1: Show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent.

Construct Truth Table

Example 2: Show that $\neg(p \wedge q)$ and $\neg p \vee \neg q$ are logically equivalent.

Construct Truth Table

Example 3: Show that $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent. (This is known as the **conditional-disjunction equivalence**.).

Construct Truth Table

Example 3 (Solution)

From the truth table, we can say a conditional statement is true if its hypothesis is false or its conclusion is true

Hence $\neg p \vee q$

p	q	$\neg p \vee q$
T	T	T
T	F	F
F	T	T
F	F	T

ANOTHER WAY TO VERIFY EQUIVALENCE

We remember some important equivalences

We use them to verify any given equivalence

IMPORTANT EQUIVALENCE LAWS

Law	Explanation
Identity	False AND True is always false True OR False is always true
Domination	True OR anything is always true False AND anything is always false
Idempotent	Telling a statement many times is no different from telling it once
Negation	A true statement and another false statement
Double negation	যক্ষা হইলে রক্ষা নাই, এই কথার ভিত্তি নাই
Absorption	

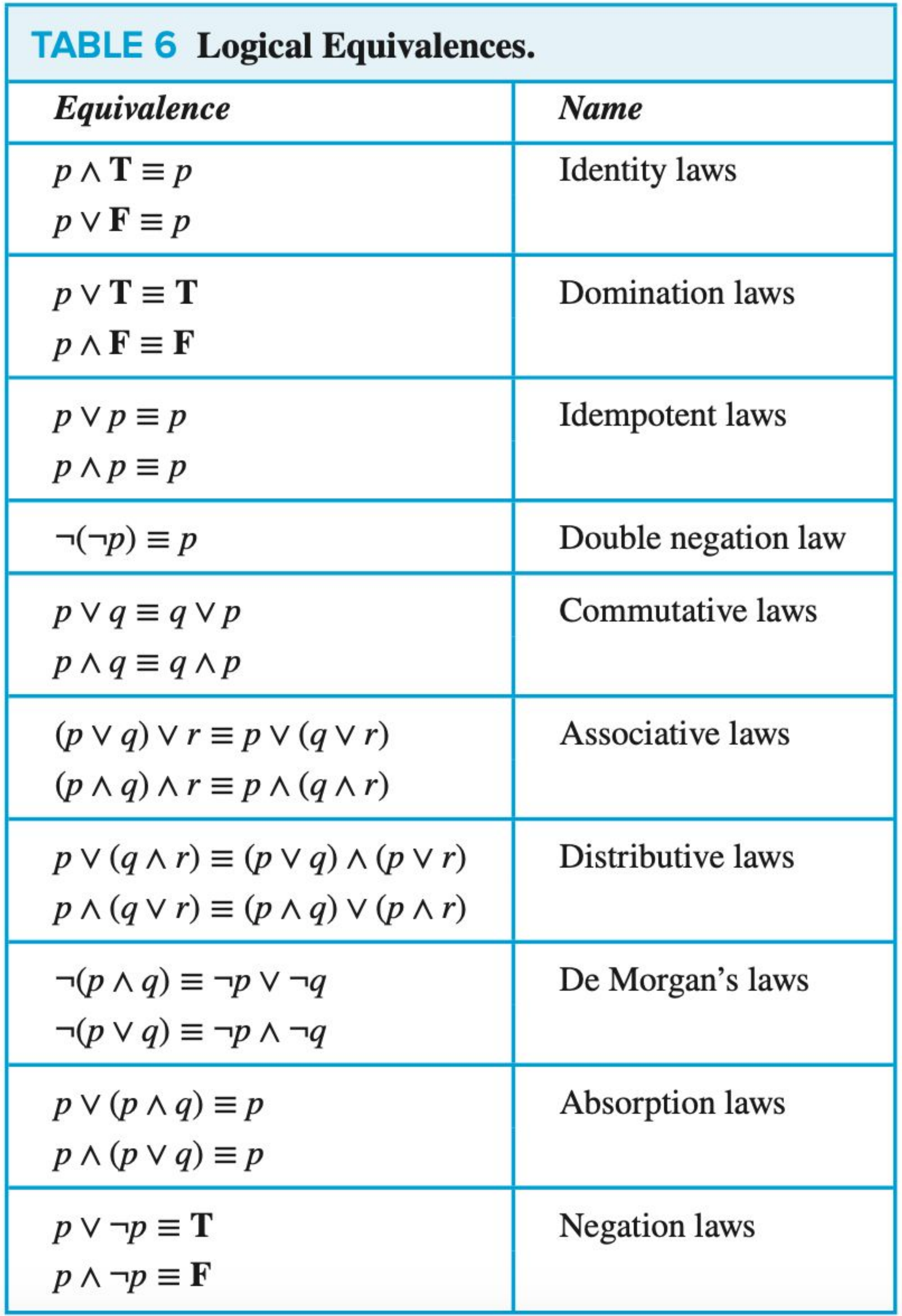


TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \land \mathbf{T} \equiv p$ $p \lor \mathbf{F} \equiv p$	Identity laws
$p \lor \mathbf{T} \equiv \mathbf{T}$ $p \land \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \lor p \equiv p$ $p \land p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \lor q \equiv q \lor p$ $p \land q \equiv q \land p$	Commutative laws
$(p \lor q) \lor r \equiv p \lor (q \lor r)$ $(p \land q) \land r \equiv p \land (q \land r)$	Associative laws
$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$ $p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$	Distributive laws
$\neg(p \land q) \equiv \neg p \lor \neg q$ $\neg(p \lor q) \equiv \neg p \land \neg q$	De Morgan’s laws
$p \lor (p \land q) \equiv p$ $p \land (p \lor q) \equiv p$	Absorption laws
$p \lor \neg p \equiv \mathbf{T}$ $p \land \neg p \equiv \mathbf{F}$	Negation laws

IMPORTANT EQUIVALENCE LAWS

Same operators

Different operators

Law	Explanation
Commutative	You may rewrite a proposition by switching the statements
Associative	Which operator you are binding first – it doesn't matter
Distributive	…
De Morgan's Law	

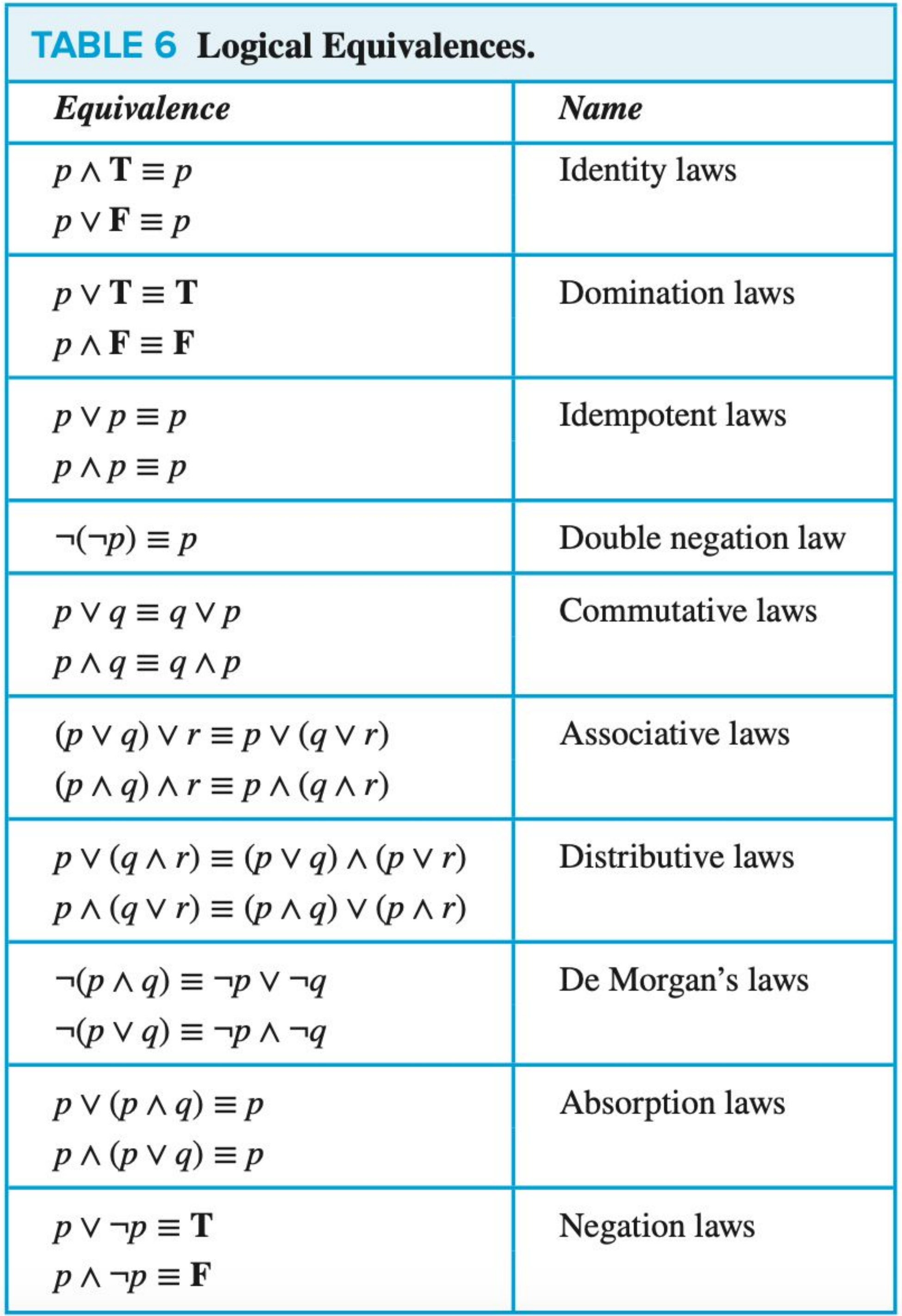


TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

IMPORTANT EQUIVALENCE LAWS

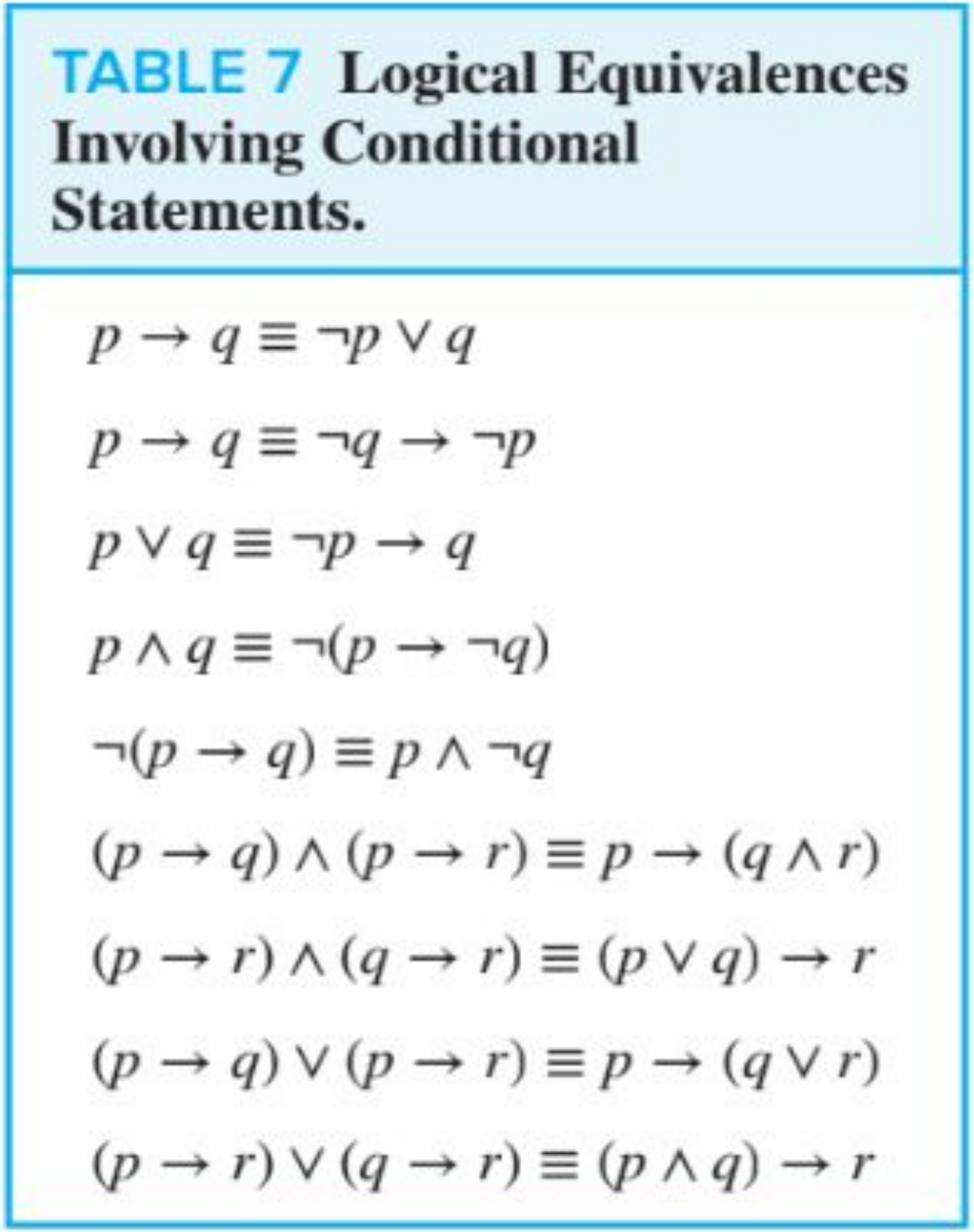


TABLE 7 Logical Equivalences Involving Conditional Statements.

$$p \to q \equiv \neg p \vee q$$

$$p \to q \equiv \neg q \to \neg p$$

$$p \vee q \equiv \neg p \to q$$

$$p \wedge q \equiv \neg(p \to \neg q)$$

$$\neg(p \to q) \equiv p \wedge \neg q$$

$$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$$

$$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$$

$$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$$

$$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$$

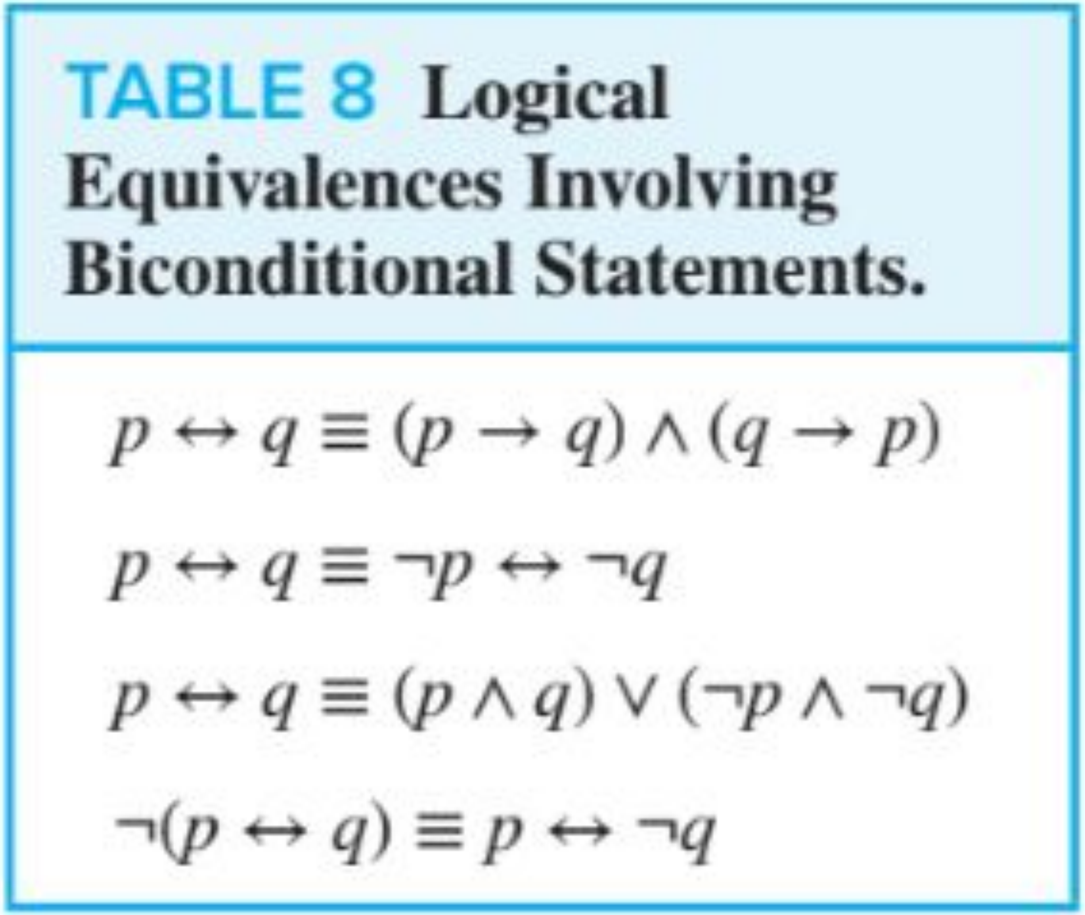


TABLE 8 Logical Equivalences Involving Biconditional Statements.

$$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$$

$$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$$

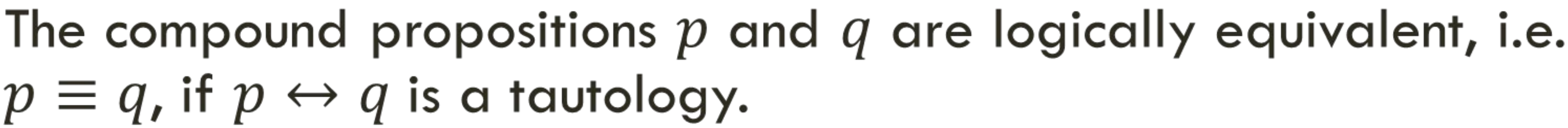
$$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$$

$$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$$

QUICK EXERCISE

Prove that $(p \to q) \land (p \to r) \equiv p \to (q \land r)$

FORMAL DEFINITION OF EQUIVALENCE



The compound propositions p and q are logically equivalent, i.e. $p \equiv q$, if $p \leftrightarrow q$ is a tautology.

Assignment No. 1

Home Work No. 1 to 7

- Slide No. 68 - 75
- Due Date: December 01, 2024 (12.30 p.m. - 1.50 p.m.)
- Submission Location: Room No. 325
- Handwritten and Individual
- Include your Name, Student ID and Course Information.

Home Work No. 1

Prove that $\neg p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \vee r)$

Home Work No. 2

Prove that $\big((p \rightarrow q) \land (q \rightarrow r)\big) \rightarrow (p \rightarrow r)$ is a tautology

Home Work No. 3

Prove that $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Home Work No. 4

PROBLEMS OF USING TRUTH TABLE FOR VERIFYING EQUIVALENCE

Prove or disprove the following equivalence using truth table:

$(p \vee q) \wedge (r \vee s)$

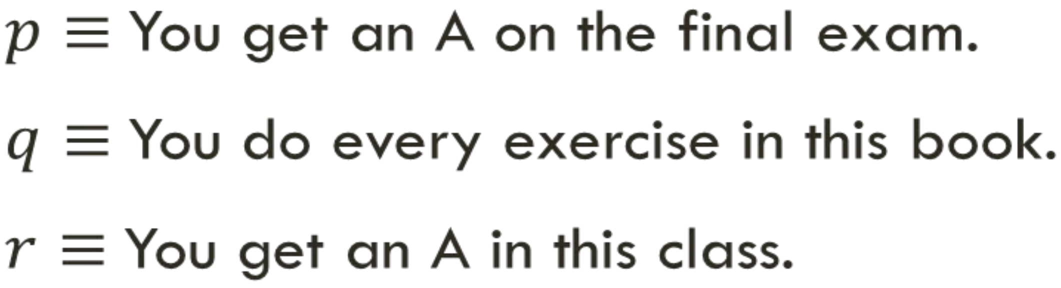
$\equiv (p \wedge r) \vee (q \wedge r) \vee (p \wedge s) \vee (q \wedge s)$

Home Work No. 5

Using truth table, prove or disprove the following equivalence:

$$(p \land q) \to r \equiv (p \to r) \land (q \to r)$$

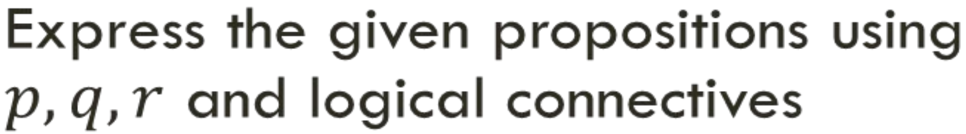
Home Work No. 6



$p \equiv$ You get an A on the final exam.

$q \equiv$ You do every exercise in this book.

$r \equiv$ You get an A in this class.



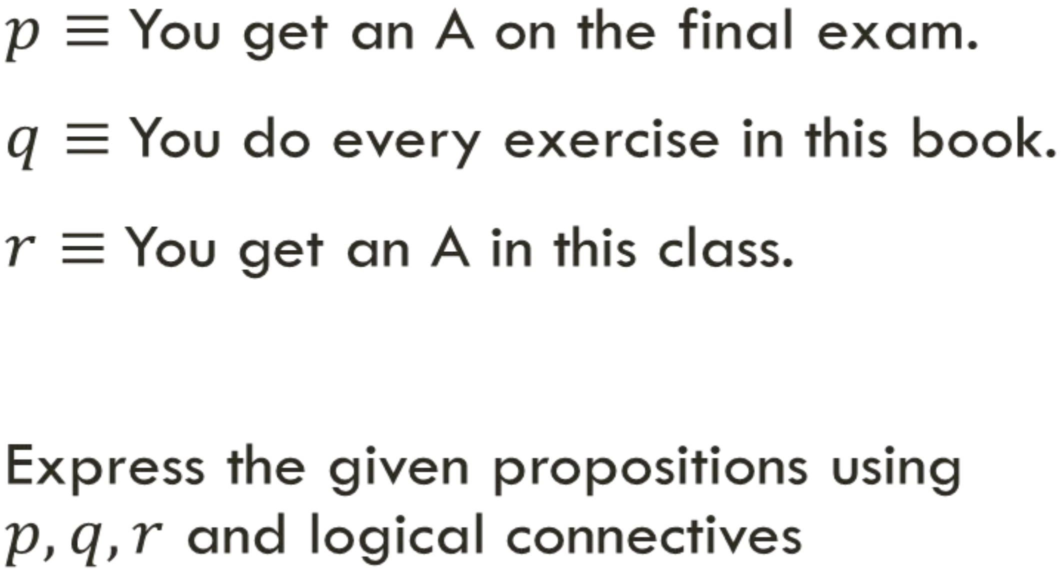
Express the given propositions using p, q, r and logical connectives

You get an A in this class, but you do not do every exercise in this book.

You get an A on the final, you do every exercise in this book, and you get an A in this class.

To get an A in this class, it is necessary for you to get an A on the final.

PROBLEM



$p \equiv$ You get an A on the final exam.

$q \equiv$ You do every exercise in this book.

$r \equiv$ You get an A in this class.

Express the given propositions using p, q, r and logical connectives

You get an A on the final, but you don’t do every exercise in this book; nevertheless, you get an A in this class.

Getting an A on the final and doing every exercise in this book is sufficient for getting an A in this class.

You will get an A in this class if and only if you either do every exercise in this book or you get an A on the final.

Home Work No. 7

Find out if the following statements are true or false:

If 2+2=4, elephants fly in the sky.

If 2+2=5, elephants fly in the sky.

If a pigeon can fly in the sky, then 2+2=4.

If a chicken can fly in the sky, then 2+2=5.

If UIU Mars Rover team is in the top 10, then elephants fly in the sky.