Department of Computer Science
Ashoka University

CS-1110 Discrete Mathematics - Spring 2025

QUESTION BANK

Instructors	:	Prof. Partha Pratim Das
Teaching Fellow	:	Shubhajit Dey

DISCLAIMER

This document consists almost all the problems expressed through the likes of Practice Problem Sets, Flash Quizzes, Quizzes and Programming Assignments, during the span of the **CS1110 - Discrete Mathematics** course in the Spring 2025 semester at Ashoka University.

Although the course was designed to be a gateway course, i.e., intended to lay the basic mathematical foundations of an undergraduate CS major, nevertheless a significant programming component was also incorporated in the syllabus unlike any traditional undergraduate course on Discrete Mathematics. The goal was to cultivate and strengthen a proper programming attitude amongst the students. **C** progarmming language was used as the primary language and all relevant code files can be found at this GitHub repo.

Further the ultimate section of this document, named **Extras**, contains some sort of hints/solutions to seleted questions from the indicated Problem Sets. The answers are not crisp/rigorous in any 'absolute' sense, rather they have been designed to guide the students to approach proof wrting in an appropriate manner. Some proofs have been left *incomplete*, with scope of formal expansion, so that this document does not end up becoming an academic opium for undergrads.

Any instance of typos and conflicting content is requested to be reported over an email to shubhajit.acad[@]icloud[dot]com.

INTENTION SEPARATES PEOPLE. SMALL ACTIONS. SMALL CHOICES. THE SMALL EFFORT THAT MEANS EVERYTHING. DOING THE RIGHT THING FOR THE RIGHT REASON IS INCREDIBLY RARE. IT'S GIVING TO GIVE, INSTEAD OF GIVING TO GET. IT'S DOING THE WORK, FOR THE WORK ITSELF. IT'S CARING WITHOUT CONSTRAINT, WITHOUT EXPECTATION, WITHOUT AN AGENDA. INTENTION CAN'T BE FAKED. AND IT CAN'T BE BEAT.

Contents

MODULE 01 - MODELS & PROOFS

1.1 Problem Set 01 - Propositional logic

1. Use $\neg$, $\rightarrow$, $\vee$ and $\wedge$ to express the following declarative sentences in propositional logic; in each case state what your respective propositional variables p, q, etc, mean:

   (a) If the sun shines today, then it won't shine tomorrow

   (b) If Bob has installed central heating, then he has sold his car, or he has not paid his mortgage.

   (c) Today it will rain or shine, but not both

2. Consider the following situation and an argument for it.

   Situation: Reason about whether a given number n is prime.
   Argument:
   #1: If n is not divisible by any number other than 1 and itself, then n is a prime number.
   #2: n is divisible by 1 and itself only.
   Therefore,
   #3: n is a prime number.

   Introduce propositional variables, represent the entire argument as a semantic entailment relation, and show that it holds true.

3. Consider the following situation and an argument for it.

   Situation: You are debugging a program and want to conclude that the input file format is correct.
   Argument:
   #1: If there is an error in the input file format and the error-checking module is disabled, the program crashes.
   #2: The program did not crash.
   #3: The error-checking module was disabled.
   Therefore,
   #4: The input file format is correct.

   Introduce propositional variables, represent the entire argument as a semantic entailment relation, and show that it holds true.

4. Construct a truth table for each of these propositional formulas. Be mindful of the precedence of logical connectives, as it may affect the evaluation. Refer to the slides for the correct precedence order.

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

   (b) $p_1 \wedge \neg p_2 \leftrightarrow p_3 \vee p_4$

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

5. Compute the truth table for $p \to q \to r$. Unsure about the order of evaluation? Should it be $(p \to q) \to r$ or $p \to (q \to r)$? Compute the truth table for both orders of evaluation and compare the results to see if they yield the same truth table.

   Do the same for the following formulas as well: $p \lor q \lor r$ and $p \land q \land r$.

6. Construct a truth table for each of these propositional formulas:

   (a) $(q \to \neg p) \lor (\neg p \to \neg q)$

   (b) $(p \lor \neg t) \land (p \lor \neg s)$

   (c) $(((p \land \neg q) \to r) \to (\neg r \to (p \to q)))$

   (d) $(p \lor q) \to (p \oplus q)$ [Refer to slides for the connective "$\oplus$"]

   (e) $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow \neg r)$ [Refer to slides for the connective "$\leftrightarrow$"]

7. Let p, q and r be the propositional variables such that:

p	You get an A on the final exam
q	You do every exercise in this book
r	You get an A in this class

   Write the following declarative statements (propositions) using p, q, and r and logical connectives

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

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

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

   (d) 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

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

8. For the semantic entailment proofs provided below, determine which ones hold and which ones do not.

   (a) $p \to q, s \to t \models p \lor s \to q \land t$

   (b) $p \lor q, \neg q \lor r \models p \lor r$

   (c) $p \to (q \lor r), \neg q, \neg r \models \neg p$

   (d) $q \to r \models (p \to q) \to (p \to r)$

9. $\boxed{\text{Definition}}$ Let ϕ and ψ be propositional formula. We say that ϕ and ψ are semantically equivalent ($\equiv$) iff $\phi \models \psi$ and $\psi \models \phi$ hold.

   Show the semantic equivalence of the following formulas:

$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	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
$p \rightarrow q \equiv \neg p \vee q$ $p \rightarrow q \equiv \neg q \rightarrow\neq p$ $p \vee q \equiv \neg p \rightarrow q$ $p \wedge q \equiv \neg(q \rightarrow \neg p)$ $(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$ $(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$ $(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$ $(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$ $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$	Equivalences involving $\rightarrow$

10. Alice and Bob have each written an algorithm for a function that takes two sorted lists, List1 and List2, of lengths m and n, respectively, and merges them into a third list, List3. Part of Alice'c code and the corresponding part of Bob's code are given below:

Alice

```
1  if ((i + j ≤ m + n) && (i ≤ m) &&
   ((j > n) || (List1[i] ≤ List2[j])))
2      List3[k] = List1[i]
3      i = i + 1
4  else
5      List3[k] = List2[j]
6      j = j + 1
7  k = k + 1
```

Bob

```
1  if (((i+j ≤ m+n) && (i ≤ m) && (j > n)) ||
   ((i + j ≤ m + n) && (i ≤ m) &&(List1[i] ≤
   List2[j])))
2      List3[k] = List1[i]
3      i = i + 1
4  else
5      List3[k] = List2[j]
6      j = j + 1
7  k = k + 1
```

Do these parts of the code do the same thing ? Notice that both the codes are exactly the same except for **line 1** (assuming, both have used the same local variables)

Using the propositional variables given below, express line1 in both codes as a propositional formula. Then, demonstrate that they achieve the same result.

p to stand for $i + j \leq m + n$
q to stand for $i \leq m$
r to stand for $j > n$, and
s to stand for List1$[i] \leq$ List2$[j]$

1.2 Problem Set 02 - Predicate logic

1. Formalise the following arguments and then use Natural Deduction to infer the conclusion from the given premises. Do not forget to clearly mention the rules of inference you use in each step.

   (a) **Premises :** "All dementors are fierce."; "Some dementors do not drink coffee."
   Conclusion : "Some fierce creatures do not drink coffee."

   (b) **Premises :** "All crows are richly coloured."; "No large bird live on honey."; "Birds that do not live on honey are dull in color."
   Conclusion : "Crows are small."

2. Prove the following logical arguments using Natural Deduction. State the rules of inference used at each step.

   (a)

$$\begin{array}{l} \forall x\ (P(x) \rightarrow \neg Q(x)) \\ \exists x\ (Q(x) \wedge R(x)) \\ \hline \therefore \exists x\ (\neg P(x) \wedge Q(x)) \end{array}$$

   (b)

$$\begin{array}{l} (\exists x\ P(x)) \rightarrow (\forall x\ \neg Q(x)) \\ \hline \therefore (\exists x\ Q(x)) \rightarrow (\forall x\ \neg P(x)) \end{array}$$

3. Mention which variables are free and bound in the following statements.

   (a) $\exists x\ P(x) \vee \exists x\ Q(y) \wedge \exists y Q(x)$
   (b) $\forall y\ \exists x\ P(x) \vee P(y)$
   (c) $\forall y\ \exists x\ (P(x) \vee Q(y))$
   (d) $\forall x\ P(x) \rightarrow \exists y\ Q(x)$

4. Negate the following encoded mathematical statement using concepts seen in the module for predicate logic. Which concept is being primarily used here?

$$\forall \epsilon > 0 \Big(\exists \delta > 0 \big(\forall x \in \mathbb{R} \ni (|x - c| < \delta \Rightarrow |f(x) - f(c)| < \epsilon) \big) \Big).$$

5. The TAs of this Discrete Mathematics course have been suspected of unauthorised access to the AC02-216 lab at Ashoka. They have made the following statements to PPD. Rudransh said, Elvia did it." Monu said, "I did not do it." Elvia said, "Vedant did it." Vedant said, "Elvia lied when she said that I did it." $\star$

   (a) If professor knows that exactly one of them is lying, who did it?
   (b) If professor knows that exactly one of them is telling the truth, who did it?

   Explain your reasoning.

6. Consider the following predicates. Using these predicates (and these predicates only), write a predicate that describes the following relationship $N(x, y)$: "x is the nephew of y's spouse". ⋆⋆

   $P(x, y)$: "x is a parent of y"
   $S(x, y)$: "x and y are siblings"
   $M(x, y)$: "x is married to y"
   $F(x)$: "x is a female"

1.3 Problem Set 03 - Propositional logic-II

1. Consider the following argument. Demonstrate its validity by encoding it in propositional logic as $\phi_1, \phi_2, \phi_3 \vdash \psi$ and providing a proof of its correctness.

   ▶ Alice is from Vienna
   ▶ It isn't the case that both Alice and Bob are Viennese
   ▶ The same goes for Alice and Russell: they aren't both from Vienna
   Therefore,
   ▶ Both Bob and Russell are not Viennese

2. Consider the following argument. Demonstrate its validity by encoding it in propositional logic sequent and providing a proof of its correctness.

   ▶ It isn't true that Alice is a logician while Bob isn't
   ▶ Also, it isn't the case that both Eve and Bob are logicians
   Therefore,
   ▶ It isn't true that both Alice and Eve are logicians

3. Prove the validity of the following sequents:

   (a) $(p \land q) \land r, s \land t \vdash q \land s$

   (b) $q \to (p \to r), \neg r, q \vdash \neg p$

   (c) $(p \to r) \land (q \to r) \vdash p \land q \to r$

   (d) $p \to q, r \to s \vdash p \land r \to q \land s$

4. Prove the validity of the following well-known derived rules:

   (a) $\boxed{\text{Hypothetical Syllogism}}$ $\dfrac{p\to q \quad q\to r}{p\to r}$ HS

   (b) $\boxed{\text{Disjunctive Syllogism}}$ $\dfrac{p\lor q \quad \neg p}{q}$ DS

   (c) $\boxed{\text{Resolution}}$ $\dfrac{p\lor q \quad \neg p\lor r}{q\lor r}$ DS

5. Consider the following argument. Demonstrate its validity by encoding it in propositional logic sequent and providing a proof of its correctness.

   ▶ Discrete Mathematics is not tough and it is a gate course in CS
   ▶ We will attend Discrete Mathematics classes only if it is tough. That is, We will attend Discrete Mathematics classes means that it is tough
   ▶ If we do not attend Discrete Mathematics classes, then we will stage a play
   ▶ If we stage a play, then we will have fun
   Therefore,
   ▶ We will have fun

6. Consider the following argument. Demonstrate its validity by encoding it in propositional logic sequent and providing a proof of its correctness.

   ▶ If Virat Kohli scores a century, then he will win the match of the match
   ▶ If Virat Kohli does not scores a century, then he will be dropped from India team
   ▶ If Virat Kohli is dropped from India team, then he will become a commentator
   Therefore,
   ▶ If Virat Kohli does not win the match of the match, then he will become a commentator.

7. Alice and Bob are two individuals, each of whom is either a knight or a knave. It is known that knights always tell the truth, while knaves always lie. Alice states: "At least one of us is a knave." Determine the identities of Alice and Bob.

   Answer. We introduce the following propositional variables to represent Alice's and Bob's identities:

p	Alice is knave
q	Bob is knave

   Alice states - at least one of us is a knave. This is given to us as premise. While the part "at least one of us is a knave" can be encoded as $(p \vee q)$, we must encode the whole statement "Alice states: at least one of us is a knave". Since Alice can be either a knave or a knight, we must account for both possibilities in our premise. Thus, the valid encoding is as follows:

$$\phi = (p \to \neg(p \vee q)) \wedge (\neg p \to (p \vee q))$$

   The above is correct because if Alice is a knave, her statement must be false, and if she is a knight, her statement must be true.

   With the premise correctly encoded, our task is to determine the exact identities of Alice and Bob. There are four possible conclusions:: $(p \wedge q)$ or $(\neg p \wedge \neg q)$ or $(\neg p \wedge q)$ or $(p \wedge \neg q)$. All that remains is to determine which conclusion follows from the premise.

   We can verify this using semantic entailment by constructing a truth table.

p	q	$p \vee q$	$\neg(p \vee q)$	$p \to \neg(p \vee q)$	$(\neg p \to (p \vee q))$	ϕ
F	F	F	T	T	F	F
F	T	T	F	T	T	T
T	F	T	F	F	T	F
T	T	T	F	F	T	F

   Thus, the validity of premise leaves us with the choice that $(p, q) = (F, T)$.
   Basically, Alice is a knight and Bob is a knave. *qed.*

8. Provide a valid construction history for each of the following propositional formulas, adhering to the precedence rules.

   (a) $(p \to q) \to \neg r \vee (q \wedge p \to r)$
   Answer:

   $$
   \begin{aligned}
   & p,\ q,\ r \\
   & (p \to q) \\
   & q \wedge p \\
   & (q \wedge p \to r) \\
   & \neg r \\
   & \neg r \vee (q \wedge p \to r) \\
   & (p \to q) \to \neg r \vee (q \wedge p \to r)
   \end{aligned}
   $$

   (b) $p \to \neg q \vee r \to p \vee s$
   Answer:

   $$
   \begin{aligned}
   & p,\ q,\ r,\ s \\
   & \neg q \\
   & \neg q \vee r \\
   & p \vee s \\
   & \neg q \vee r \to p \vee s \\
   & p \to \neg q \vee r \to p \vee s
   \end{aligned}
   $$

   (c) $p \wedge \neg q \to \neg p$
   (d) $(p \to \neg q \vee (p \wedge r) \to s) \vee \neg r$

9. Suppose a propositional formula ψ does not follow from the given set of premises $\phi_1, \phi_2, \ldots, \phi_n$. If you were to prove this, which approach would you prefer: $\phi_1, \phi_2, \ldots, \phi_n \nvdash \psi$ or $\phi_1, \phi_2, \ldots, \phi_n \nvDash \psi$? Justify your choice. ⋆

10. The table below represents the complete truth table for some propositional formula ϕ involving three propositional variables: p, q and r.

p	q	r	ϕ
F	F	F	T
F	F	T	T
F	T	F	F
F	T	T	T
T	F	F	F
T	F	T	T
T	T	F	F
T	T	T	T

   (a) Construct a disjunctive clause D_1 such that D_1 evaluates to F at the valuation (F, T, F) and evaluates to T for all other valuations.
   (b) Similarly, construct disjunctive clauses D_2 and D_3 that evaluate to F at the valuations (T, F, F) and (T, T, F) respectively, while evaluation to T otherwise.
   (c) Finally, show that $\phi = D_1 \wedge D_2 \wedge D_3$

MODULE 02 - PROOF TECHNIQUES

2.1 Problem Set 04 - Proofs

2.1.1 Part - A

1. We call a number p prime if it has exactly two factors. Suppose we define $Three-Prime$ as those numbers which have exactly three factors. Derive the general form of a $Three-Prime$ and **prove** that all $Three-Prime$s must be of your derived form.

2. The prime numbers p and q are called *twin primes* if $|p-q|=2$. Let p and q be primes. **Prove** that $pq+1$ is a square if and only if p and q are twin primes.

3. Let $F_0, F_1, F_2, \ldots F_n$ denotes the the Fibonacci sequence given by $F_0=0, F_1=F_2=1$ and satisfying $F_{n+2}=F_{n+1}+F_n$ for $n \geq 0$. We call F_n the n^{th} *Fibonacci number*. Answer the following based on this.

   (a) Show that $\gcd(F_n, F_{n+1})=1, \forall n \geq 1$.
   (b) Prove **Cassini's Identity**: $F_{n-1} \cdot F_{n+1}-(F_n)^2=(-1)^n, \forall n \geq 2$.
   (c) Let $\alpha=\frac{1+\sqrt{5}}{2}$ and $\beta=\frac{1-\sqrt{5}}{2}$. Show that

   $$F_n=\frac{\alpha^n-\beta^n}{\sqrt{5}}, \qquad \forall n \geq 0.$$

4. Choose a proof method of your choice and prove the following. State the method(s) you used to seal the deal.

   (a) If n is an even integer, then n^2 is an even integer.
   (b) If $m \in \mathbb{N}$ and $n \in \mathbb{N}$ are both perfect squares, then nm is also a perfect square. $p \in \mathbb{N}$ is a perfect square if there exists $q \in \mathbb{N}$ such that $p=q^2$.
   (c) If $n=ab$, where a and b are positive integers, then $a \geq n^{1/2}$ or $b \geq n^{1/2}$.
   (d) The sum of two rational numbers is rational.
   (e) If n is a perfect square, then $n+2$ is not a perfect square.

5. Prove the following using Principles of Mathematical Induction.

   (a) $\sum_{k=1}^{n} k=\frac{1}{2} \cdot n(n+1)$.
   (b) $\sum_{k=1}^{n} k^2=\frac{1}{6} \cdot n(n+1)(2n+1)$.
   (c) $3^{n/3}>2^{n/2}, \ \forall n \in \mathbb{N}$.

6. Prove that $a^{2^n}-1$ is divisible by 4×2^n for all odd integers a, and for all integers $n \in \mathbb{N}$. ⋆⋆

2.1.2 Part - B

1. Provide a proof by **contraposition** for the following statements.

   (a) If n is an integer and n^2 is even, then n is even.
   (b) Let $a \geq 0$. If for every $\epsilon>0$, we have $0 \leq a<\epsilon$, then $a=0$.
   (c) If m, n are natural numbers such that $m+n \geq 40$, then either $m \geq 20$ or $n \geq 20$.

2. Provide a proof by **contradiction** for the following statements.

   (a) Let $a > 0$ be a real number. If $a > 0$, then $\frac{1}{a} > 0$.
   (b) There are infinitely many prime numbers. $\star$

3. Provide a proof using mathematical induction for the following statements.

   (a) Let $a \in \mathbb{R} \setminus \{1\}$. For all $n \geq 1$, $\sum_{k=0}^{n} a^k = \frac{1-a^{k+1}}{1-a}$.
   (b) Let $a, b \in \mathbb{N}$ be distinct. For all $n \geq 1$, $(a-b)$ divides $(a^n - b^n)$.
   (c) For all $n \neq 1$ and for all $a_1, \ldots, a_n \in \mathbb{R}$, we have the AM-GM inequality:

   $$\frac{a_1 + \cdots + a_n}{n} \geq (a_1 \cdots a_n)^{1/n}.$$

   (d) For an $a(\neq 0) \in \mathbb{R}$, if $(a + \frac{1}{a}) \in \mathbb{Z}$, then $a^n + \frac{1}{a^n} \in \mathbb{Z}$ for all $n \geq 1$. $\star$
   (e) Let S be a set such that $|S| = n$. Prove that $|\mathcal{P}(S)| = 2^n$. $\star$
   (f) Show that if n is a positive integer, then $\sum_{\emptyset \neq I \subseteq \{1,\ldots,n\}} \frac{1}{\prod_{i \in I} i} = n$. $\star$
   (g) Let $x > -1$ be a real number. Prove that $(1+x)^n \geq 1 + nx$ for all $n \in \mathbb{N}$.

MODULE 03 - MATHEMATICAL STRUCTURES

3.1 Problem Set 05 - Sets

3.1.1 Part - A

1. Prove the following set identities.
   (a) $A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$
   (b) $A \cap (B \cup C) = (A \cap B) \cup (A \cup C)$
2. Let A and B be sets. Show that
   (a) $(A \cap B) \subseteq A$
   (b) $A \subseteq (A \cup B)$
   (c) $A \setminus B \subseteq A$
   (d) $A \cap (B \setminus A) = \phi$
   (e) $A \cup (B \setminus A) = A \cup B$
   (f) $(B \setminus A) \cup (C \setminus A) = (B \cup C) \setminus A$
3. Let $A, B \subseteq \Omega$. Show that $(A \cap B) \cup (A \cap \bar{B}) = A$, where $\bar{B} = \Omega \setminus B$.
4. The symmetric difference of A and B, denoted by $A \Delta B$, is the set containing those elements in either A or B, but not both A and B. Clearly, $A \Delta B = (A \cup B) \setminus (A \cap B)$.
5. What can you say about the sets A and B if $A \Delta B = A$? $\star$
6. Suppose $\Omega = \{a_1, a_2, a_3, a_4, a_5\}$. Express each of the following sets with binary strings of length 5 where the ith bit (left to right) in the string is 1 if i is in the set and 0 otherwise.
   (a) $\{a_1, a_3, a_5\}$
   (b) $\{a_1, a_3, a_4, a_5\}$
   (c) ϕ
7. Suppose $\Omega = \{a_1, a_2, a_3, a_4, a_5\}$. Determine the sets specified by the following three strings: $01100, 01010$ and $(01100) \vee (01010) = (0 \vee 0, 1 \vee 1, 1 \vee 0, 0 \vee 1, 0 \vee 0)$.
8. Suppose $\Omega = \{a_1, a_2, a_3, a_4, a_5\}$. Determine the sets specified by the following three strings: $01100, 01010$ and $(01100) \wedge (01010) = (0 \wedge 0, 1 \wedge 1, 1 \wedge 0, 0 \wedge 1, 0 \wedge 0)$.
9. Show that $|A \cup B| = |A| + |B| - |A \cap B|$.
10. We know the **De Morgan's Law** for the case of two sets. Now we attempt to prove the generalized De Morgan's Law. Prove the following where A^c is the complement of a set A: $\star$
   (a) $\bigcup_{i=1}^{n} A_i = (\bigcap_{i=1}^{n} A_i^c)^c$. (b) $\bigcap_{i=1}^{n} A_i = (\bigcup_{i=1}^{n} A_i^c)^c$.

3.1.2 Black coffee for neurons

```
This is just to present an example as to how to write robust proofs.
It looks intimidating at the first look, but clearly it is simple after
a logical break-down.
```

Definition 3.1.1. ***(Chain.)*** *For any poset $R = (S, \preceq)$, a chain is a sequence,*

$$a_1 \preceq a_2 \preceq \cdots \preceq a_n$$

where $a_i \neq a_j, \forall i \neq j$ such that each item is comparable to the next one in the chain and is smaller with respect to $\preceq$.

Definition 3.1.2. ***(Antichain.)*** *An antichain in a poset is a set of elements such that no two elements in the set are comparable. Or for any distinct x, y in an antichain set, we have $x \npreceq y$ and $y \npreceq x$.*

Theorem 3.1.3. *If the largest chain in a partial order on a set A is of size t, then A can be partitioned into t antichains.*

1. Prove the Dilworth's Lemma stated below using the above given facts.

   (Dilworth's Lemma.) For all $t > 0$, every partially ordered set with n elements must have either a chain of size at least t or an antichain of size at least $\frac{n}{t}$.

   Proof. Assume that Dilworth's lemma is false, that is $\exists$ a poset $R = (S, \preceq)$ with $t > 0$ where all chains are of size $< t$ and all antichains are of size strictly less than $\frac{n}{t}$. Consider the smallest such t, with the poset's largest chain being size $t - 1$. Theorem(3.1.3) is used to find $t - 1$ antichains that partition the set. Now, using the fact that chains must have size strictly less than $\frac{n}{t}$ by our assumption and that the antichains A_i form a partition, their sizes must sum to the size of S.

   $$\sum_{i=1}^{t-1} |A_i| \leq \sum_{i=1}^{t-1} \frac{n}{t} = \frac{n(t-1)}{t} < n. \qquad \perp$$

   Since all antichains form a partition of the set, sum of the sizes of all the antichains must exactly be n. A contradiction and hence the assumption of Dilworth's lemma to be false is incorrect .

   This completes the proof.

   qed.

3.2 Problem Set 06 - Functions

1. Why is $f : \mathbb{R} \to \mathbb{R}$ is not a function if

   (a) $f(x) = 1/x$ (b) $f(x) = -\sqrt{x}$

2. Determine whether each of these function $f : \mathbb{Z} \to \mathbb{Z}$ is injective

   (a) $f(n) = n - 1$
   (b) $f(n) = n^2 + 1$
   (c) $f(n) = n^3$
   (d) $f(n) = \lceil n/2 \rceil$, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

3. Determine whether the function $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ is onto?

   (a) $f(m, n) = 2m - n$ (b) $f(m, n) = m^2 - n^2$

4. Give an explicit formula for a function $f : \mathbb{Z} \to \mathbb{N}$ such that it is

   (a) injective, but not surjective
   (b) surjective, but not injective
   (c) bijective
   (d) neither injective not surjective

5. Let $f : A \to B$ and $g : B \to C$ be functions.

   (a) Show that if $g \circ f$ is injective, then f is injective.
   (b) Show that if $g \circ f$ is surjective, then g is surjective

6. Let $f : A \to B$ be a function. Let $E, F \subseteq A$. Then prove the following:

   (a) $f(E \cup F) = f(E) \cup f(F)$ (b) $f(E \cap F) \subseteq f(E) \cap f(F)$

7. Consider a function $f : A \to B$. Let S and T be subsets of B. Show that

   (a) $f^{-1}(S \cup T) = f^{-1}(S) \cup f^{-1}(T)$
   (b) $f^{-1}(S \cap T) = f^{-1}(S) \cap f^{-1}(T)$

8. Let $A = \{-1, 0, 2, 4, 7\}$. Find $f(A)$ whenever f is defined as:

   (a) $f(x) = \lceil x/5 \rceil$ (b) $f(x) = \lfloor (x^2 + 1)/3 \rfloor$

9. Let $L = \{0, 1\}^n$, the set of all binary strings of length exactly n. Define a function $f : \{0, 1\}^n \to \mathbb{N}$ such that $\sum_{w \in L} f(w) = 2^n$. $\star$

10. Let $L = \{0, 1\}^n$, the set of all binary strings of length exactly n. Define a function $f : \{0, 1\}^n \to \mathbb{N}$ such that $\sum_{w \in L} f(w) = 2^{n-1}$. $\star$

11. Determine the number of functions $f : \{1, 2, \ldots, 7\} \to \{1, 2, \ldots, 7\}$ such that $f(x) \neq x$ for all x. $*$

12. Determine the total number of functions $f : A \to B$ that can be defined when $|A| = m$ and $|B| = n$. $*$

3.3 Problem Set 07 - Mixed Bag

1. Consider any relation R on a set S. We define R^{-1} as the inverse relation of R. $(a,b) \in R^{-1} \leftrightarrow (b,a) \in R$. Show the following :

   (a) $(R^{-1})^{-1} = R$.

   (b) Let T be another relation on S. Show the De Morgan's Law for relations, that is, $(R \circ T)^{-1} = T^{-1} \circ R^{-1}$.

   (c) $R^{-1} \circ R$ is symmetric and reflexive over some subset of S.

2. Show the following are equivalence relations.

   (a) Let two sets A, B be related if there exists functions $f : A \to B$, $g : B \to A$ such that f, g are injections.

   (b) R is a relation over $\mathbb{Z}$, defined by $R = \{(a,b) : a, b \in \mathbb{Z} \wedge n|(a-b)\}$ for some fixed $n \in \mathbb{Z}, n \neq 0$.

3. For matrix $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, prove that for any $n \in \mathbb{N}$, $A^n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}$.

4. For $C = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, prove that for any $n \in \mathbb{N}$, $C^n = \begin{pmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{pmatrix}$, where F_n is the n^{th} Fibonacci number. Further also prove that the determinant of C^n is $(-1)^n$. $\star$

5. A recurrence of the form $f(n) = h(n)f(n-1) + g(n)$ is called a first-order linear recurrence. Determine the general solution (closed-form of f) when $h(n)$ is a constant, say r. $\star$

6. Find a closed-form representation using unrolling technique, for the solution to the recurrence
$$f(n) = \begin{cases} b & \text{if } n = 0 \\ rf(n-1) + a & \text{if } n \geq 1 \end{cases},$$
   where r and a are constants.

7. Show that the closed-form expression for the linear-order recurrence $f(n) = 4f(n-1) + 2^n$, with $f(0) = 3$ is $4^{n+1} - 2^n$. $\star\star$

8. Show that the closed-form expression for the linear-order recurrence $f(n) = 3f(n-1) + n$, with $f(0) = 10$ is $\frac{43}{4}3^n - \frac{n+1}{2} - \frac{1}{4}$.
   [Hint: for any real number $a \neq 1$, $\sum_{i=1}^{n} ia^i = \frac{n \cdot a^{n+2} - (n+1) \cdot a^{n+1} + a}{(1-a)^2}$]

9. The recurrence given in question 6 is first-order linear recurrence. With $r \neq 1$, show that $f(n) = r^n b + a\frac{1-r^n}{1-r}$.

10. Consider the following function f. For every $n \in \mathbb{N}$, $f(n)$ be number of functions one can define with domain as the set $\{1, 2, \ldots, n\}$ and codomain as the set $\{1, 2, \ldots, m\}$. Give a recursive description of f. $\star\star\star$

MODULE 04 - GROWTH OF FUNCTIONS

4.1 Problem Set 08 - Asymptotics-I

Formal Notations of Asymptotics

Consider the following definitions for functions $f : \mathbb{Z}^+ \to \mathbb{R}^+$ and $g : \mathbb{Z}^+ \to \mathbb{R}^+$.

Notation	.	Definition	Example
		Tight Bounds	
$O(g(n))$	:	$\exists c > 0, \exists n_0 > 0$ s.t. $\forall n > n_0\ f(n) \le c.g(n)$	$f(n) = 5n^2 + 2n - 17 \in O(n^2), O(n^3); \notin O(n)$
$\Omega(g(n))$	:	$\exists c > 0, \exists n_0 > 0$ s.t. $\forall n > n_0\ f(n) \ge c.g(n)$	$f(n) = 5n^2 + 2n - 17 \in \Omega(n^2), \Omega(n); \notin \Omega(n^3)$
$\Theta(g(n))$	:	$\exists c_1 > 0, \exists c_2 > 0, \exists n_0 > 0$ s.t. $\forall n > n_0$ $c_1.g(n) \le f(n) \le c_2.g(n)$	$f(n) = 5n^2 + 2n - 17 \in \Theta(n^2), \Theta(n^3); \notin \Theta(n)$
		Loose Bounds	
$o(g(n))$	:	$\forall c > 0, \exists n_0 > 0$ s.t. $\forall n > n_0\ \lvert f(n)\rvert \le c.\lvert g(n)\rvert$	$f(n) = 7n - 17 \in o(n^2)$
$\omega(g(n))$	:	$\forall c > 0, \exists n_0 > 0$ s.t. $\forall n > n_0\ \lvert f(n)\rvert \ge c.\lvert g(n)\rvert$	$f(n) = 7n - 17 \in \omega(1)$
		Approximation	
$\sim (g(n))$	:	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$	$f(n) = n(n-1) \sim n^2$

1. Find the $\Theta-$bounds for the following recurrences.

   (a) $T(n) = 4T(n/2) + c$ where $T(1) = c_0$

   (b) $T(n) = T(n/4) + T(n/2) + n \cdot c$ where $T(1) = c_0$

   (c) $T(n) = T(n-2) + T(n-4)$, where $T(0) = T(1) = T(2) = T(3) = c_0$.

   (d) $T(n) = T(n-1) + T(n-2) + k$, where $T(0) = 0$ and $T(1) = 1$.

2. Solve the following linear-homogeneous recurrences and comment on their $O-$bounds.

   (a) $F(n) = 7F(n-1) - 12F(n-2),\ n \ge 2$ and $F(0) = 5, F(1) = -5$.

   (b) $F(n) = F(n-1) + 2F(n-2),\ n \ge 3$ and $F(1) = 0, F(2) = 6$.

   (c) $F(n) = -F(n-1) + 4F(n-2) + 4F(n-3),\ n \ge 3$ and $F(0) = 8, F(1) = 6, F(2) = 26$.

   (d) $F(n) = 4F(n-1) - 4F(n-2),\ n \ge 3$ and $F(1) = 1, F(2) = 3$.

   (e) $F(n) = 8F(n-1) - 16F(n-4),\ n \ge 4$ and $F(0) = 1, F(1) = 4, F(2) = 28, F(3) = 32$.

   (f) $F(n) = -3F(n-1) - 3F(n-2) - F(n-3),\ n \ge 3$ and $F(0) = 1, F(1) = -2, F(2) = -1$.

3. For function f, g and h mapping from $\mathbb{N}$ to $\mathbb{R}^+$, prove the following:

   (a) If $f = O(h)$ and $g = O(h)$, then $f + g = O(h)$, where $f + g : \mathbb{N} \to \mathbb{R}^+$ defined as $\big(f + g\big)(x) = f(x) + g(x)$.

(b) If $f = O(h)$ and $g = O(h)$, then $f \cdot g = O(h)$, where $f \cdot g : \mathbb{N} \to \mathbb{R}^+$ defined as $\big(f \cdot g\big)(x) = f(x)g(x)$.

4. For $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, show that $p(x) = O(x^n)$.

5. Prove the following asymptotic identities for $n \in \mathbb{N}$:

   (a) $n! = O(n^n)$.

   (b) $\log n! = O(n \log n)$.

   (c) $\ln n = O(n)$.

6. Prove that for some $k \in \mathbb{N}$, $\sum_{i=1}^{k} i^8 = O(k^9)$. $\star$

7. Prove that for some $k \in \mathbb{N}$, $\sum_{i=1}^{k} i^8 = \Omega(k^9)$. $\star\star\star$

8. Prove that $n^2 + 17n = n^2 + o(n \ln n)$. $\star\star\star$

9. Prove the result that $n^2 = o(n^2 \ln n)$. As a corollary to this result, can we show that $n^2 = O(n^2 \ln n)$? $\star\star\star$

4.2 Problem Set 09 - Asymptotics-II

1. Is it true that $f = O(g)$ implies $\log f = O(\log g)$?
2. Show that if $f(n) = \log n!$, then $f(n)$ is $O(n \log n)$.

   [Hint: $n! = n \cdot (n-1) \cdots 2 \cdot 1 \leq n \cdot n \cdots n \cdot n$]
3. Show that if $f(n) = \frac{n^2+1}{n+1}$, then $f(n)$ is $O(n)$. Provide explicit values for c and n_0.

   [Hint: $\frac{n^2+1}{n+1} \leq \frac{n^2+n^2}{n+1} \leq \frac{2n^2}{n} = 2n$]
4. Show that if $f(n) = n^{2.5}$, then $f(n)$ is $2^{O(\log n)}$.
5. A function $f(n)$ is said to have a quasi-linear growth rate if $f(n) = \Theta(n \log n)$. Show that if $f(n) = n \log n - 10n + 3$, then $f(n)$ exhibits quasi-linear growth.
6. Show that if $f(n) = \sqrt{n} - \log n$, then $f(n)$ exhibits sub-linear growth.
7. Show that if $f(n) = logn - \log\log n + 2$, then $f(n)$ exhibits sub-linear growth.
8. Show that if $f(n) = 1 + 2 + \cdots + n$, then $f(n)$ is $\Omega(n^2)$. Provide explicit values for c and n_0.
9. Suppose $f(n)$ is $O(g(n))$. Does it follow that $2^{f(n)}$ is $O(2^{g(n)})$?
10. Show that if $f(n) = 2n^3 + 4n^2 \log n$, then $f(n)$ is $O(n^3)$.
11. Show that, for $0 \leq a < b$, $n^a = o(n^b)$.
12. Show that, for $a > 0$, $\log n = o(n^a)$.
13. Show that if $f(n)$ is $o(g(n))$, then $f(n)$ cannot be $\Omega(g(n))$.
14. Show that if $f(n) = \frac{n}{\log n}$, then $f(n)$ is $O(n)$ but it is not $\Omega(n)$.
15. Show the following:

    (a) $\sqrt{n}$ is $o(n)$.

    (b) n is $o(n \log\log n)$.

    (c) $n \log n$ is $o(n^2)$.

MODULE 05 - THEORY OF DISCRETE PROBABILITY

5.1 Problem Set 10 - Probability distributions-I

1. Consider $\mathbb{P} : \Omega \to [0, 1]$ to be a probability distribution on a non-empty sample space Ω. For events $A, B \subseteq \Omega$ prove the **identities**:

   (a) $\mathbb{P}(\bar{A}) = 1 - \mathbb{P}(A)$.

   (b) $\mathbb{P}(A \cap \bar{B}) = \mathbb{P}(A) - \mathbb{P}(A \cap B)$.

2. Consider the **single-coin tossing experiment** with sample space $\Omega = \{H, T\}$. Give a probability distribution explicitly for the followign situtaions:

   (a) The coin is biased and after running the experiment sufficiently large number of times it is establishes that H appears thrice as many times as T does.

   (b) The coins are biased and it is statistically established that for every three appearances of H, we get two appearances to T.

3. For a fixed $n \in \mathbb{N}$, let $\Omega = \{0, 1\}^n$, the set of all n-length binary strings. Further any $\omega \in \Omega$ can be represented as $\omega = b_1 b_2 b_3 \ldots b_n$, where each b_i is either 0 or 1.
   Let $p, q \in \mathbb{R}$ such that $0 \leq p, q \leq 1$ and $p+q = 1$. Show that the function $\mathbb{P} : \Omega \to [0, 1]$ defined as:
   $$\mathbb{P}(\omega) = \prod_{i=1}^{n} p^{b_i} \cdot q^{1-b_i},$$
   is a probability distribution on Ω. ⋆⋆

4. Let $\Omega = \{0, 1\}^4$ be the set of all binary strings of length 4. Consider the **experiment of generating binary strings of length 4** which realises the uniform probability distribution. Assume appearances of 0s and 1s to be independant and answer the following:

   (a) What is the probability that the generated binary string will contain at least two consecutive 0s, given the fact that the binary string starts with 0?

   (b) What is the probability that the generated binary string will contain at least two consecutive 0s, given the fact that the binary string starts with 1?

5. Consider $\mathbb{P} : \Omega \to [0, 1]$ to be a probability distribution on a non-empty sample space Ω. Answer the following using the fact that a probability distribution distributes over disjoint events.

   (a) For events $A, B \subseteq \Omega$, show that $\mathbb{P}(\bar{A}|B) = 1 - \mathbb{P}(A|B)$.

   (b) For events $A, B \subseteq \Omega$, it is given that $\mathbb{P}(A) = 1/5$, $\mathbb{P}(A|B) = 1/3$ and $\mathbb{P}(B|A) = 1/7$. Calculate $\mathbb{P}(B)$.

6. For $\Omega = \{1, 2, 3, 4\}$, consider $\mathbb{P} : \Omega \times \Omega \to [0, 1]$ to be the uniform probability distribution on $\Omega \times \Omega$. Let $A, B \subseteq \Omega \times \Omega$ be the event where $A := \{(s, t) \in \Omega \times \Omega \mid s + t = 6\}$ and $B := \{(s, t) \in \Omega \times \Omega \mid s + t = 0 \ mod \ 2\}$. Compute $\mathbb{P}_B(A)$.

7. Consider $\mathbb{P} : \Omega \to [0, 1]$ to be a probability distribution on a non-empty sample space Ω. For independant events $A, B \in \Omega$ and event $C \in \Omega$ show that:

$$\mathbb{P}(C) \cdot \mathbb{P}(A \cap B) \leq \mathbb{P}(A) \cdot \mathbb{P}(B).$$

8. Let $\Omega = \{0, 1\}^4$ be the set of all binary strings of length 4. Consider the **experiment of generating binary strings of length 4** which realises the uniform probability distribution. Let $A \subseteq \Omega$ be the event that the generated binary string starts with 1 and $B \subseteq \Omega$ be the event that the generated binary string contains even number of 1s. Under the assumption that appearances of 0s and 1s are independant, determine whether A and B are independant or not.

9. Consider $\mathbb{P} : \Omega \to [0, 1]$ to be a probability distribution on a non-empty sample space Ω. If $A, B, C \subseteq \Omega$ such that $A \cap \bar{C} = B \cap \bar{C}$, then show that $|\mathbb{P}(A) - \mathbb{P}(B)| \leq \mathbb{P}(C)$. $\star$

10. Consider $\mathbb{P} : \Omega \to [0, 1]$ to be a probability distribution on a non-empty sample space Ω. Show that if $A, B \subseteq \Omega$ are independant events, then events $\bar{A} \subseteq \Omega$ and B are also independant.

11. Consider probability distribution $\mathbb{P}_1 : \Omega \to [0, 1]$ and $\mathbb{P}_2 : \Omega \to [0, 1]$ on a non-empty sample space Ω, show that

$$\sum_{\omega \in \Omega} |\mathbb{P}_1(\omega) - \mathbb{P}_2(\omega)| \leq 2.$$

► HINT: Triangle inequality: $|a + b| \leq |a| + |b|$, might come handy.

5.2 Problem Set 11 - Probability distributions-II

1. Let events $A,\ B,\ C \subseteq \Omega$ form a *cover* of the sample space Ω. If $\mathbb{P}(A) = \mathbb{P}(B) = \mathbb{P}(C)$ and these events are mutually independant then compute $\mathbb{P}(A \cap B \cap C)$. Consider $\mathbb{P}$ realises the uniform probability distribution.

2. Let A and B be two mutually disjoint events. Further, let A and B together be independent of event C. If $\mathbb{P}[A] + P[B] = a$ and $\mathbb{P}[C] = b$, and $\mathbb{P}$ realises the uniform probability distribution then compute $\mathbb{P}[A \cup B \cup C]$. ⋆

   ▶ HINT : Use the Principle of Inclusion-Exclusion, i.e., $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|$.

3. Let $A,\ B$ and C be three events such that the events A and B are mutually disjoint events. Further it is given that $\mathbb{P}[A \cup B] = 1$, $\mathbb{P}[A \cap C] = 1/4$ and $\mathbb{P}[C] = 7/12$. Compute $\mathbb{P}[B \cap C]$. ⋆

4. Let $A,\ B,\ C$ be any three events with $\mathbb{P}[A] = 0.3$, $\mathbb{P}_A[B] = 0.2$, $\mathbb{P}_A[C] = 0.1$ and $P_A[(B \cap C)] = 0.05$. Then compute $P[A \setminus (B \cup C)]$.

5. Consider a random variable X such that $\mathbb{E}[X^2] = 10$ and $\mathbb{E}[X]^2 = 6$. Compute $\mathbb{E}[(X - \mathbb{E}[X])^2]$.

6. **Joint distribution.** Suppose Π be a random experiment with sample space $\Omega = \{(a, b) \mid 1 \leq a, b \leq 4\}$ and it realises the uniform probability distribution P on Ω. Consider the following two random variables $X,\ Y$ from Ω to the set $\{1, 2, 3, 4\}$ defined as:
$$X(a, b) = \max(a, b);\ Y(a, b) = \min(a, b) \qquad \forall (a, b) \in \Omega.$$
   Now, consider the joint random variable (X, Y). Clearly,
   $(X, Y) : \Omega \times \Omega \to \{1, 2, 3, 4\} \times \{1, 2, 3, 4\}$. Compute $\mathbb{P}_{X \times Y}\big((3, 2)\big)$.

7. Consider the joint random variable (X, Y) with domain $\{0, 1\} \times \{0, 1, 2\}$. It is given $\mathbb{P}_{(X,Y)}[(a, b)] = \frac{a+b}{9},\ \forall (a, b) \in \{0, 1\} \times \{0, 1, 2\}$.

   (a) **Marginal distributions.** Determine marginal distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$.

   (b) **Independence of distributions.** Are random variables X and Y independent?

8. Let $X : \Omega \to S$ and $Y : \Omega \to T$ be two random variables such that the joint random varible (X, Y) realises the uniform distribution $S \times T$. Determine the distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$, qualitatively. Further, are random variables X and Y independent?

9. Let $\mathbb{P}$: $\{0, 1\}^3 \to [0, 1]$ be the Uniform probability distribution. Let $X : \{0, 1\}^3 \to \{0, 1\}$ be a random variable that maps elements $(b_1 b_2 b_1) \in \{0, 1\}^3$ to 1 if and only if $b_1 = b_3$. Let $Y : \{0, 1\}^3 \to 0, 1$ be a random variable that maps elements $(b_1 b_2 b_1) \in \{0, 1\}^3$ to 1 if and only if $(b_1 + b_2 + b_3) = 2$. Compute $\mathbb{P}_{(X,Y)}(1, 1)$. ⋆

10. Let X be a random variable with the probability distribution
    $\mathbb{P}_X : \{-2, -1, 0, 1, 2\} \to [0, 1]$ defined as $\mathbb{P}_X(x) = k(1 + |x|)^2$, where $k \in \mathbb{R}$ is a constant. Compute $\mathbb{P}_X(0)$.

11. Let X be a random variable such that $Range(X) = \{0, 1, 2, \ldots, n\}$. Then show that $\sum_{i=1}^{n} \mathbb{P}_X[X \geq i] = E[X]$. ⋆

12. Consider Π, the experiment of rolling two unbiased dice. Let X and Y be random variables where X encodes sum of the two faces and Y encodes the absolute value of the difference of the two faces. Show that $E[XY]$ is 0.

    ▶ HINT: Use linearity of expectations, i.e., $E[X+Y] = E[X] + E[Y]$.

MODULE 06 - THEORY OF GRAPHS

6.1 Problem Set 12 - Graphs-I

To test command over basic definitions & notations

1. Consider a *directed* graph G. Prove that graph G being strongly connected implies that G is weakly connected, and not the other way around.

2. Consider a *directed* graph G. Prove that graph G being a Null graph implies that G is also an Empty graph. Provide a counter-example to show that the other way around is not always true.

3. The general graph shown in the following figure(1) goes by the name of *GraphBuster* in standard literature. Count and determine the cardinality of V and E in *GraphBuster*.

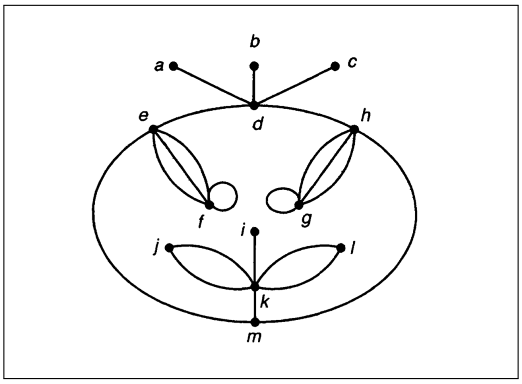


Figure 1: GraphBuster

4. A graph of order n is called *complete*, denoted by K_n provided that each pair of distinct vertices forms an edge. Show that a complete graph of order n has $n(n-1)/2$ edges.

Problems of type $\neg$ (simple)

5. Let G be a general graph. Show that the sum of the degrees of all the vertices of G is an even number, and consequently, the number of vertices of G with odd degree is even.

6. If G is a simple graph of order $n \geq 3$, such that for all pairs of distinct vertices x and y in G that are not adjacent, we have $deg(x) + deg(y) \geq n$, then show that G must be connected. ⋆

7. Let G be the graph such that elements of $\{1, 2, 3, \cdots, 20\}$ form its vertices. In G, two vertices (integers) are joined by an edge if and only if their difference is an odd integer. Show that G is a bipartite graph.

8. Prove that if a multigraph G is bipartite, then each of its cycles has even length. *Note that: length of any cycle/path is the number of edges it is composed of.*

9. For a fixed $n \in \mathbb{N}$, let $\mathrm{G_n}$ be the graph such that elements of $\{0, 1\}^n$, the set of all $n-$length binary strings, form its vertices. In $\mathrm{G_n}$ any two vertices are joined by an edge if and only if they differ in exactly one 1-bit. Show that G is a bipartite graph. ⋆

10. Prove that a graph of order n with at least

$$\frac{(n-2)(n-1)}{2} + 1$$

edges must be connected.

11. Prove that a graph of order n with every vertex having degree at least $\frac{n}{2}$ must be connected.

12. In a simple graph if two vertices x and y are joined by a path then, show that they are also joined by a simple path.

6.2 Problem Set 13 - Graphs-II

Problems of type (simple)

1. The two graphs shown below in figure (2) have the same number of vertices and edges. Prove that despite these they are not isomorphic.

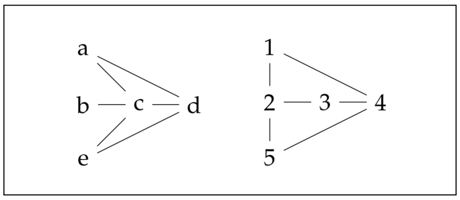


Figure 2: Sample graphs 01

2. Consider the two graphs shown below in figure (3), where both the graphs have same degree sequence $(3, 3, 3, 3, 3, 3)$. Show that despite this, they are not isomorphic.

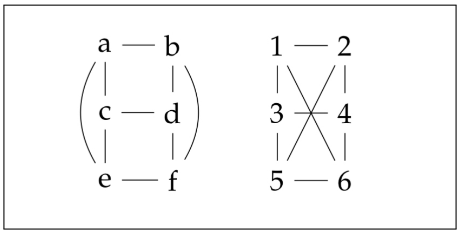


Figure 3: Sample graphs 02

3. A graph has 26 vertices and 58 edges. There are five vertices of degree 4, six vertices of degree 5, and seven vertices of degree 6. If the remaining vertices all have the same degree, what is this degree?

4. A graph has 24 vertices and 30 edges. It has five vertices of degree 4, seven pendant vertices, and seven vertices of degree 2. All other vertices have degree 3 or 4. How many vertices of degree 4 are there?

Problems of type $\neg$ (simple)

5. Prove or Disprove, whether a bipartite graph can have K_3 as its subgraph?

6. Let $d_0(G)$ be the least among the degrees of the vertices of an $n-$vertex graph G. Prove that if $d_0(G) \geq (n-1)/2$, then the graph G is connected. $\star$

7. Prove that a graph G with v vertices and e edges has at least $v - e$ connected components. [Hint : use induction on e.]

8. Prove that a connected graph G with n vertices contains at least $n - 1$ edges. [Hint : the proof might be an application of the result in the previous question, when proved!]

9. If G is a connected graph with v vertices and e edges, then $v \leq e + 1$.

10. If G is a connected graph, then removing an edge from a cycle will not make G a disconnected graph. $\star$

6.3 Problem Set 14 - Graphs-III

1. A graph $G = (V, E)$ is called k-regular if $\mathsf{deg}(v) = k$ for all $v \in V$. A graph is called regular if it is k-regular for some k. Give example of a regular bipartite graph.
2. Prove that every induced subgraph of a complete graph is complete.
3. Prove that every subgraph of a bipartite graph is bipartite.
4. If two graphs G_1 and G_2 are isomorphic then their degree sequences are the same.
5. What is the sum of the entries in a row of the adjacency matrix of an undirected simple graph?
6. Let u, v, and w be three distinct vertices in a graph. There is a path between u and v and also there is a path between v and w. Prove that there is a path between u and w.
7. Suppose $(d_1, \ldots, d_n)$ be a degree sequence of a tree. Determine $\sum_{i=1}^{n} d_i$.
8. Show that the number of vertices n in a binary tree is always odd.
9. Let p be the number of pendant vertices in a binary tree T with n vertices. Show that
$$p = \frac{n+1}{2}.$$
10. Let $k \in \mathbb{N}$ be the height of a binary tree T. Determine the maximum number of leaf nodes of T.
11. Consider the graph defined by the adjacency matrix provided below.
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 1 & 0 \\ & & & 0 & 0 & 1 & 0 & 1 \\ & & & & 0 & 0 & 1 & 1 \\ & & & & & 0 & 1 & 1 \\ & & & & & & 0 & 1 \\ & & & & & & & 0 \end{bmatrix}$$
    (a) Determine if it is an Euler graph.
    (b) Determine if it admits a Hamiltonian circuit.
    (c) Give a spanning tree of this graph.

ASSESSMENTS

7.1 Flash Quiz 01

1. Code the following two statements in propositional logic and then find the *negation*, *converse*, *inverse* and *contrapositive* for each statement if applicable. If for any of the statements the *negation*, *converse*, *inverse* or *contrapositive* is / are not possible to apply to the question, mention the same. [**2 * 5 = 10**]

   (a) *To be in Gryffindor, it suffices that Harry is courageous.*
   (b) *Every student of CS-1110-01 is rational.*

   Part (a)
   p: Harry is courageous.
   q: Harry is in Gryffindor.
   Formal Statement: $p \rightarrow q \equiv (\neg p \vee q)$
   Negation: $\neg(\neg p \vee q) \equiv p \wedge \neg q$
   Converse: $q \rightarrow p$
   Inverse: $\neg p \rightarrow \neg q$
   Contrapositive: $\neg q \rightarrow \neg p$

   Part (b)
   P(x) : x is a student of CS-1110-01.
   Q(x) : x is a rational.
   Formal Statement: $\forall x(P(x) \rightarrow Q(x)) \equiv \forall x(Q(x) \vee \neg P(x))$
   Negation: $\neg(\forall x(Q(x) \vee \neg P(x)) \equiv \exists x(\neg Q(x) \wedge P(x))$
   Converse: $\forall x(Q(x) \rightarrow P(x))$
   Inverse: $\forall x(\neg P(x) \rightarrow \neg Q(x))$
   Contrapositive: $\forall x(\neg Q(x) \rightarrow \neg P(x))$

2. Formalise the following argument and use both Semantic Entailment and Resolution to show its validity: [**4 + 3 + 3 = 10**]

   - **Premises**
     (a) *It is not wintry or Rita has her jacket*;
     (b) *Rita does not have her jacket or she does not catch a cold*;
     (c) *It is wintry or Rita does not catch a cold.*
   - **Conclusion**: *Rita does not catch a cold*

   Formalisation - 4 marks
   p: it is wintry
   q: Rita has her jacket
   r: Rita catches a cold
   premise 1: $\neg p \vee q$
   premise 2: $\neg q \vee \neg r$
   premise 3: $p \vee \neg r$
   conclusion: $\neg r$

   Sematic Entailment - 3 marks
   The complete truth table must consist of each of the three propositional variables, all

three premises and the conclusion. The truth table must contain eight distinct rows.

Resolution - 3 marks

$$\dfrac{\dfrac{(\neg p \lor q) \qquad (p \lor \neg r)}{\therefore (q \lor \neg r)}\ \text{(RES)} \qquad (\neg q \lor \neg r)}{\therefore \neg r}\ \text{(RES)}$$

which is “Rita does not catch a cold”.

3. Formalize the following using the given coded atomic statements. Justify the steps for your final answer. [**10**]

For walking on the path to be safe, it is necessary but not sufficient that grapes not be ripe along the path and for foxes not to have been seen in the area.

w: the walk along the path is safe.
f: there are foxes in the area.
g: the grapes are ripe.

The only correct answer: $\Big[w \to (\neg f \land \neg g)\Big] \land \neg\Big[(\neg f \land \neg g) \to w\Big]$
Reasoning: If seen as $[\phi] \land \neg[\psi]$, then the clause ϕ takes care of the ‘necessary’ condition and the clause ψ takes care of the ‘not sufficient’ condition imposed. Rest of the formalisation is starightforward.

7.2 Flash Quiz 02

1. Consider the recursive function $T(n) = 4T(n/2) + 3$ where $T(1) = 1$.
[**5 + 10 = 15 marks**]

   (a) Unroll and find the closed form of this recursive function.

   (b) Find (with proof) a $\Theta-$bound for this function.

$$\begin{aligned} T(n) &= 4T(n/2) + c \\ &= 16T(n/4) + 4c + c \\ &= 64T(n/8) + 16c + 4c + c \\ &= 256T(n/16) + 64c + 16c + 4c + c \\ &\vdots \\ &= 4^k T(n/2^k) + c \cdot \sum_{0}^{k-1} 4^i \end{aligned}$$

   Let n be such that, $n/2^k = 1$. Then $n = 2^k$ and $k = \log_2 n$. Thus we get:
   $T(n) = 4^{\log_2 n} + c(1 + 4 + 16 + 64 + \cdots + 4^{\log_2 n - 1})$
   $\Rightarrow T(n) = n^{\log_2 4} + 3(\frac{(4^{\log_2 n} - 1)}{4-1})$
   $\Rightarrow T(n) = n^2 + 3(\frac{n^{\log_2 4} - 1}{3})$
   $\Rightarrow T(n) = n^2 + n^2 - 1$
   $\Rightarrow T(n) = 2n^2 - 1$
   Considering the given definition of $\Theta-$bound, for $c_1 = 1,\ c_2 = 3,\ n_0 = 2$ we have:

$$T(n) \in \Theta(n^2).$$

2. Show that $\log n! = O(n \log n)$. [**5 marks**] $1 \le n;\ 2 \le n;\ 3 \le n;\ 4 \le n; \cdots;\ n \le n.$
   Thus we get :
   $1 \cdot 2 \cdot 3 \cdot \cdots \cdot n \le \underbrace{n \cdot n \cdot n \cdot \cdots \cdot n}_{n\ times}$
   $\Rightarrow n! \le n^n.$
   $\Rightarrow \log n! \le n \log n.$
   Thus for $c = 1$ and $n_0 = 1$ we have $\log n! = O(n \log n)$.

7.3 Flash Quiz 03

1. Let A and B be two mutually disjoint events. Further, let A and B be events independent of event C. If $\mathbb{P}[A] + \mathbb{P}[B] = a$ and $\mathbb{P}[C] = b$, and $\mathbb{P}$ realises the uniform probability distribution then compute $\mathbb{P}[A \cup B \cup C]$. [**6 marks**]
   ANSWER : $\mathbf{a + b - ab}$.
   Let Ω be the sample space. We consider the Inclusion-Exclusion Principle:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|$$
$$\implies \frac{|A \cup B \cup C|}{|\Omega|} = \frac{|A|}{|\Omega|} + \frac{|B|}{|\Omega|} + \frac{|C|}{|\Omega|} - \frac{|A \cap B|}{|\Omega|} - \frac{|B \cap C|}{|\Omega|} - \frac{|C \cap A|}{|\Omega|} + \frac{|A \cap B \cap C|}{|\Omega|}$$

...due to $\mathbb{P}$ being the Uniform probability distribution...

$$\implies \mathbb{P}[A \cup B \cup C] = \mathbb{P}[A] + \mathbb{P}[B] + \mathbb{P}[C] - \mathbb{P}[A \cap B] - \mathbb{P}[B \cap C] - \mathbb{P}[C \cap A] + \mathbb{P}[A \cap B \cap C]$$
$$\implies \mathbb{P}[A \cup B \cup C] = a + b - 0 - ab + 0$$
$$\implies \mathbb{P}[A \cup B \cup C] = a + b - ab.$$

2. Consider probability distribution $\mathbb{P}_1 : \Omega \to [0,1]$ and $\mathbb{P}_2 : \Omega \to [0,1]$ on a non-empty sample space Ω. Show that [**4 marks**]

$$\sum_{\omega \in \Omega} |\mathbb{P}_1(\omega) - \mathbb{P}_2(\omega)| \leq 2.$$

   Due to the Triangle inequality: $|a + b| \leq |a| + |b|$, we have:

$$\begin{aligned}\sum_{\omega \in \Omega} |\mathbb{P}_1(\omega) - \mathbb{P}_2(\omega)| &\leq \sum_{\omega \in \Omega} |\mathbb{P}_1(\omega)| + |\mathbb{P}_2(\omega)| \\ &= \sum_{\omega \in \Omega} |\mathbb{P}_1(\omega)| + \sum_{\omega \in \Omega} |\mathbb{P}_2(\omega)| \\ &= 2 \qquad \text{[since each are probability distributions.]}\end{aligned}$$

   This completes the proof. *qed*

3. Consider the joint random variable (X, Y) with domain $\{0,1\} \times \{0,1,2\}$, corresponding to some random variables X and Y. It is given $\mathbb{P}_{(X,Y)}[(a,b)] = \frac{a+b}{9}$, $\forall (a,b) \in \{0,1\} \times \{0,1,2\}$.

   (a) Determine marginal distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$. [**7 marks**]
   (b) Are random variables X and Y independent? [**3 marks**]

   (a) We have $\mathbb{P}_X[x_0] = \sum_{t \in \{0,1,2\}} \mathbb{P}_{(X,Y)}[(x_0, t)]$ and $\mathbb{P}_Y[y_0] = \sum_{s \in \{0,1\}} \mathbb{P}_{(X,Y)}[(s, y_0)]$. Thus,

   - $\mathbb{P}_X[0] = \frac{0+0}{9} + \frac{0+1}{9} + \frac{0+2}{9} = 1/3$.
   - $\mathbb{P}_X[1] = \frac{1+0}{9} + \frac{1+1}{9} + \frac{1+2}{9} = 2/3$.
   - $\mathbb{P}_Y[0] = \frac{0+0}{9} + \frac{1+0}{9} = 1/9$.
   - $\mathbb{P}_Y[1] = \frac{0+1}{9} + \frac{1+1}{9} = 1/3$.
   - $\mathbb{P}_Y[2] = \frac{0+2}{9} + \frac{1+2}{9} = 5/9$.

   (b) **No.** If random variable X and Y are independant, than we have

$$\mathbb{P}_{(X,Y)}[(a,b)] = \mathbb{P}_X[a] \times \mathbb{P}_Y[b], \qquad \forall (a,b) \in \{0,1\} \times \{0,1,2\}.$$

   Here, $\mathbb{P}_{(X,Y)}[(0,0)] \neq \mathbb{P}_X[0] \times \mathbb{P}_Y[0]$. Thus, X and Y are not independant.

7.4 Flash Quiz 04

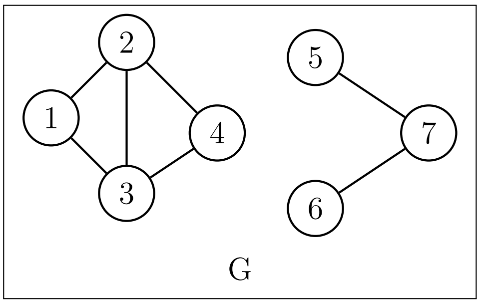
1. Proof that if a simple undirected graph G is Euler, then any degree sequence of G must be comprised of even natural numbers only. [**8 marks**]

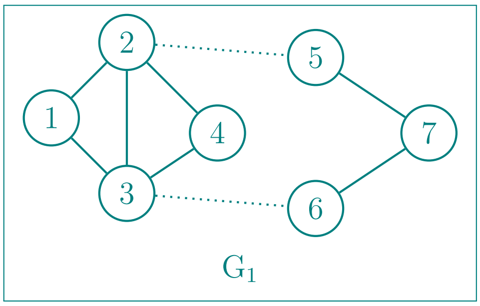
   Proof. Since G is Euler, let one of its Eulerian cycle be of the following form:

   $$[x_0 \xrightarrow{e_0} x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_2} \dots \xrightarrow{e_{n-1}} x_n \xrightarrow{e_n} x_0].$$

   with possible repetition amongst the vertices x_i, but not amongst the edges e_i. Arbitrarily, pick a vertex x. For this vertex x, there must be two distinct edges $e_1 = \{a, x\}$ and $e_2 = \{y, b\}$. Hence, for each vertex x, if there is one edge *incident* on it, then there is one edge to *outgoing* from it too. Each such pairing introduces 2 to $\deg(x)$, so $\deg(x)$ is even. Since vertex x was chosen arbitrarily, the proof is complete. $\square$

2. Consider the disconnected graph G given the following figure. Draw the graph G_1 after adding the minimum number of edges in G so that the result becomes a simple Euler Graph. Justify your answer. [**3 marks**]





   We use the result we have proved in Question 1 to identify that vertices $2, 3, 5, 6$ are the only vertices with odd degrees. Further, for a graph to be Euler, it is necessary that it is connected. These leads to the above given G_1. **Observe that, one can also come up with an existential proof, nevertheless in all scenarios G_1 remains unique!**

3. Solve the following recurrence relations and write their closed forms where $n \in \mathbb{N}$:

   (a) $a_n = 6a_{n-1} - 9a_{n-2}$, $\quad \forall n \geq 2$. Given $a_0 = 1, a_1 = 6$. [**3 marks**]
   $a_n = (1 + n) \cdot 3^n$.
   The characterisitic equation in this case is : $r^2 - 6r + 9 = 0$.

   (b) $a_n = 7a_{n-1} - 12a_{n-2}$, $\quad \forall n \geq 2$. Given $a_0 = 5, a_1 = -5$. [**3 marks**]
   $a_n = 25 \cdot 3^n - 20 \cdot 4^n$.
   The characterisitic equation in this case is : $r^2 - 7r + 12 = 0$.

   (c) $a_n = 4a_{n-1} - 4a_{n-2}$, $\quad \forall n \geq 3$. Given $a_1 = 1, a_2 = 3$. [**3 marks**]
   $a_n = (1 + n) \cdot 2^{n-2}$.
   The characterisitic equation in this case is : $r^2 - 4r + 4 = 0$.

7.5 Quiz 01

Question 01 **[4 + 6 = 10 marks]**

Solve the following **'Exciting Life Problem'** using **natural deduction**.

Those people who read are not stupid. Poulomi can read and is wealthy. All people who are not poor and are smart are happy. Happy people have exciting lives. Can anyone be found with an exciting life?

1. Formalize the statements into a set of logical clauses.
2. Provide a correct and complete proof by natural deduction.

Answer to Question 01

- **Set of Predicates**

$\text{READ}(x)$:	x reads
$\text{SMART}(x)$:	x is smart (also, x is not stupid)
$\text{WEALTHY}(x)$:	x is wealthy (also, x is not poor)
$\text{HAPPY}(x)$:	x is happy
$\text{EXCITING}(x)$:	x has an exciting life

1. **Formalisation** The above statements become the following when formalised.
   - $\forall x \ \big(\text{READ}(x) \rightarrow \text{SMART}(x)\big)$
   - $\big(\text{READ}(Poulomi) \wedge \text{WEALTHY}(Poulomi)\big)$
   - $\forall x \ \big(\big(\text{WEALTHY}(x) \wedge \text{SMART}(x)\big) \rightarrow \text{HAPPY}(x)\big)$
   - $\forall x \ \big(\text{HAPPY}(x) \rightarrow \text{EXCITING}(x)\big)$

2. **Natural Deduction** Each of the above formalised statements become premises. For obvious reasons this proof is not unique. **Yes, there is someone with an exciting life.**

1.	$\forall x \ \big(\text{READ}(x) \rightarrow \text{SMART}(x)\big)$	premise.
2.	$\big(\text{READ}(Poulomi) \wedge \text{WEALTHY}(Poulomi)\big)$	premise.
3.	$\forall x \ \big(\big(\text{WEALTHY}(x) \wedge \text{SMART}(x)\big) \rightarrow \text{HAPPY}(x)\big)$	premise.
4.	$\forall x \ \big(\text{HAPPY}(x) \rightarrow \text{EXCITING}(x)\big)$	premise.
5.	$\big(\text{READ}(Poulomi) \rightarrow \text{SMART}(Poulomi)\big)$	Universal Instantiation of 1.
3.	$\big(\big(\text{WEALTHY}(Poulomi) \wedge \text{SMART}(Poulomi)\big) \rightarrow \text{HAPPY}(Poulomi)\big)$	Universal Instantiation of 3.
6.	$\big(\text{HAPPY}(Poulomi) \rightarrow \text{EXCITING}(Poulomi)\big)$	Universal Instantiation of 4.
7.	$\text{WEALTHY}(Poulomi)$	Simplification or $\wedge_{e,2}$ 2.
8.	$\text{READ}(Poulomi)$	Simplification or $\wedge_{e,1}$ 2.
9.	$\text{SMART}(Poulomi)$	Modus Ponens on 5, 8.
10.	$\text{WEALTHY}(Poulomi) \wedge \text{SMART}(Poulomi)$	Conjunction or $\wedge_i$ 7, 9.
11.	$\text{HAPPY}(Poulomi)$	Modus Ponens on 3, 10.
12.	$\text{EXCITING}(Poulomi)$	Modus Ponens on 6, 11.
13.	$\exists x \ \big(\text{EXCITING}(x)\big)$ *(conclusion)*	Existantial Instatiation of 12.

– OR –

Question 02 **[5 + 5 = 10 marks]**

Solve the following mandatory sub-parts.

a. Let $\phi = \exists x(P(y,z) \wedge (\forall y(\neg Q(y,x) \vee P(y,z))))$. Consider the terms $t_1 = w$ (w is a variable), $t_2 = f(x)$ and $t_3 = g(y,z)$, where f and g are function symbols with arity 1 and 2 respectively.

   i. Compute $\phi[t_1/x]$, $\phi[t_1/y]$.

   ii. Which of the terms t_2, t_3 are free for x in ϕ.

   iii. Which of the terms t_2, t_3 are free for y in ϕ.

   [**Hint:** Scope of $\exists x$ in ϕ is $P(y,z)$]

b. Consider the predicate formula ϕ given by

$$\phi := \forall x\ (P(x) \rightarrow Q(x, f(x))),$$

where P and Q are predicates and f is a function symbol. Suppose we fix the domain of discourse or the ground set $\Omega = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Define P, Q and f such that ϕ is true.

[**Hint:** Appropriate subset of Ω for P, Q stays same; and f reverses! Palindrome - that reads the same forward and backward!]

Answer to Question 02

a. Here, we have the following logical compound statement.

$$\phi = \exists x\Big(\underbrace{P(y,z)}_{I} \wedge \big(\forall y(\neg \underbrace{Q(y,x)}_{II} \vee \underbrace{P(y,z)}_{III})\big)\Big)$$

x ... II and III are scope of the quantifier $\forall y$ in ϕ, so x is free in II.
y ... Since I is the scope of the quantifier term $\exists x$ in ϕ, y is free in I and binded in II, III.
z ... ϕ has no quantifier on z, hence z is free throughout ϕ.

   i. $\phi[t_1/x] = \exists x(P(y,z) \wedge (\forall y(\neg Q(y,w) \vee P(y,z))))$,
   $\phi[t_1/y] = \exists x(P(w,z) \wedge (\forall y(\neg Q(y,x) \vee P(y,z))))$.

   ii. $\mathbf{t_2}$ — t_2 is free for x in ϕ; t_3 is not free for x in ϕ.

   iii. $\mathbf{t_3}$ — t_2 is not free for y in ϕ; t_3 is free for y in ϕ.

b. Although the following is a natural answer, there might be other correct answers too!

$P(x)$:	$\mathbf{x} \in \{000, 010, 101, 111\}$
$Q(x,y)$:	$\mathbf{x} = \mathbf{y}$ as strings.
$f(x)$:	the bit-reversal function (since we only have 3-bit binary strings here, under **f** the first bit gets replaced by the third and the third by first bit).

Question 03 **[5 + 5 = 10 marks]**

Solve the following questions.

a. Prove using the principle of mathematical induction that $n^3 + 2n$ is divisible by 3 for all $n \in \mathbb{N}, n \geq 1$.

b. Rudransh has a stock of Rs. 2/- notes and Rs. 5/- notes, only. Prove using the principle of mathematical induction, that Rudransh can dispense any amount of money, say Rs. x, where x is a positive integer ≥ 4, using these two denominations.
[**Hint:** The statement to be proved is as follows: For all $n \geq 4$, there exists $a, b \in (\mathbb{N} \cup \{0\})$ such that $n = 2a + 5b$.]

Answer to Question 03

a. We need to prove that $\forall n \in \mathbb{N},\ P(n) : 3 \mid n^3 + 2n$.
BASE CASE: $(1)^3 + 2(1) = 3$, which clearly is divisible by 3.
INDUCTIVE HYPOTHESIS: We assume, for some $k \in \mathbb{N}$ and $k > 1,\ P(k)$ is true.
INDUCTIVE STEP: We inspect the formulation for $k + 1 \in \mathbb{N}$.

$$\begin{aligned}
(k+1)^3 + 2(k+1) &= (k+1)[(k+1)^2 + 2] \\
&= (k+1)[k^2 + 2k + 3] \\
&= (k+1)[3m + 3] && \text{[using IH, where } m \in \mathbb{N}] \\
&= 3(k+1)(m+1).
\end{aligned}$$

Therefore, for any $k \in \mathbb{N},\ P(k+1)$ holds whenever $P(k)$ is true. This completes the proof by induction. *qed.*

b. We need to prove that $\forall n \in \mathbb{N}$ *and* $n \geq 4,\ P(n) :\ \exists n_1, n_2 \in \mathbb{N} \cup \{0\} \ni n = 2n_1 + 5n_2$.
BASE CASE: Observe $P(4)$ holds where $n_1 = 2$ and $n_2 = 0$.
INDUCTIVE HYPOTHESIS: We assume, for some $k \in \mathbb{N}$ and $k > 4,\ P(k)$ is true.
INDUCTIVE STEP: We inspect the formulation for $k + 1 \in \mathbb{N}$. Due to the IH we have, $\exists m_1, m_2 \in \mathbb{N} \cup \{0\} \ni k + 1 = 2m_1 + 5m_2 + 1$.

i. CASE I : $k + 1$ is even.
Observe that any even integer greater than 4 will have at least two 2s involved in its sum. Basically any even integer greater than 4 can be written as $4 + p$, where p is an appropriate positive integer.

$$\begin{aligned}
k+1 &= 2m_1 + 5m_2 + 1 \\
&= 2(n_1 + 2) + 5m_2 + 1 \\
&= 2n_1 + 5m_2 + 5 \\
&= 2n_1 + 5(m_2 + 1) \\
&= 2n_1 + 5n_2
\end{aligned}$$

ii. CASE II : $k + 1$ is odd.
Observe that any odd integer greater than 4 will have at least one 5s involved in its sum. Basically any odd integer greater than 4 can be written as $5 + q$, where q is an appropriate positive integer.

$$\begin{aligned}
k+1 &= 2m_1 + 5m_2 + 1 \\
&= 2m_1 + 5(n_2 + 1) + 1 \\
&= 2m_1 + 5n_2 + 6 \\
&= 2(m_1 + 3) + 5n_2 \\
&= 2n_1 + 5n_2
\end{aligned}$$

Therefore, for any $k \in \mathbb{N},\ P(k+1)$ holds whenever $P(k)$ is true. This completes the proof by induction. *qed.*

Question 04 **[2 + 5 + 8 = 15 marks]**

Welcome to the **'Game of Logic'** which has the following two assumptions:

i. Logic is difficult or not many students like logic.
ii. For logic to be not difficult it is sufficient that mathematics is easy.

Formalise the above two assumptions (**2 marks**) and then answer the following questions on logical argument by considering these two assumptions to be two premises. The first and second assumption becomes `premise 1` and `premise 2` respectively.

a. Consider "Mathematics is not easy, if many students like logic", to be the conclusion and show that the argument `premise 1`, `premise 2` $\vdash$ `conclusion` is valid.
[1 + 4 = 5 marks]

b. Consider "Not many students like logic, if mathematics is not easy.", to be the conclusion and show that the argument `premise 1`, `premise 2` $\models$ `conclusion` is not valid.
[1 + 7 = 8 marks]

Answer to Question 04

▶ Formalisation

p : logic is difficult
q : mathematics is easy
r : many students like logic

a. **To prove :** $(p \lor \neg r);\ (q \to \neg p) \vdash (r \to \neg q)$.

1. $p \lor \neg r$ ... premise
2. $q \to \neg p$ ... premise
3. $\neg r \lor p$ ... Equivalent to 1.
4. $r \to p$ ... Definition of $\to$
5. $p \to \neg q$ ... Contrapositive of 2.
6. $r \to \neg q$ ... Hypothetical Syllogism on 4. & 5. [***conclusion***]

b. **To prove :** $(p \lor \neg r);\ (q \to \neg p) \not\models (\neg q \to \neg r)$.

.	p	q	r	$(p \lor \neg r)$ premise	$(q \to \neg p)$ premise	$(\neg q \to \neg r)$ conclusion
1.	F	F	F	T	T	T
2.	F	F	T	F	T	F
	⋮	⋮	⋮	⋮	⋮	⋮
6.	T	F	T	**T**	**T**	**F**

In the last row of the Truth Table above, for a certain combination of semantic values of the propositional variables, all both the premises are TRUE but the conclusion is FALSE. Hence, the proof by semantic entailment is complete. *qed.*

▶ *You can use the code in 'semantic_entailment_01.c' uploaded in the Google Classroom to verify the above proof.*

– OR –

Question 05 **[5 + 5 + 5 = 15 marks]**

Solve the following mandatory sub-parts.

a. The table below represents the complete truth table for some propositional formula ϕ involving three propositional variables: p, q and r. Provide an example of a formula ϕ in conjunctive normal form (CNF) whose truth table matches the given one.

p	q	r	ϕ
F	F	F	F
F	F	T	T
F	T	F	F
F	T	T	T
T	F	F	F
T	F	T	T
T	T	F	T
T	T	T	T

b. The following is a proof of the sequent $p \vee q, \neg q \vee r \vdash p \vee r$. Carefully examine the proof and identify the each of the inference rules used below and labelled as Rule 1 through Rule 5. If any of these rules involve an assumption, explicitly state so.

1.	$p \vee q$	... premise
2.	$\neg q \vee r$	... premise
3.	p	... Rule1
4.	$p \vee r$	... Rule2
5.	q	... Rule3
6.	$\neg\neg q$	... Rule4
7.	r	... Rule5
8.	$\vdots$	$\vdots$

[**Hint:** $\frac{p_1 \vee p_2, \quad \neg p_1}{p_2}$ DS (Disjunctive Syllogism)]

c. Consider the following statement and express it as a predicate formula using appropriate predicates. The numbers a and b should be treated as constants.

The numbers a and b are bigger than their common factors.

Answer to Question 05

a. **An answer** is $\phi = (p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (\neg p \vee q \vee r)$, this can be made compact.
Step 1 Identify rows where ϕ is False.
Step 2 To ensure ϕ is False in the rows identified above, we construct clauses that eliminate these cases. Each clause must be False in at least one of the rows where ϕ is False. For each row where ϕ is False, we construct a clause which is False only in that row. So we get clauses $(p \vee q \vee r)$, $(p \vee \neg q \vee r)$ and $(\neg p \vee q \vee r)$.
Step 3 Combine the above clauses using conjunction. This formula ensures that ϕ is False in exactly the rows where $\phi =$ False in the given truth table, and True otherwise.

b. **Rule1** is *assumption*.
Rule2 is *introduction of disjunction on 3* **or** $\vee i_1$ 3 **or** *addition on 3*.
Rule3 is *assumption*.
Rule4 is *introduction of double-negation on 5* **or** $\neg\neg i$ 5.
Rule5 is *disjunctive syllogism on 2, 6*.

c. $\mathbf{C(x)}$: x is a common factor of a and b
$\mathbf{G(x,y)}$: x is greater than y
Encoding : $\forall x\Big(C(x) \to (G(a,x)) \wedge G(b,x)\Big)$.

7.6 Quiz 02

Question 01 **[5 marks]**

Prove that: A natural number n is composite if and only if it is divisible by a natural number less than or equal to $\sqrt{n}$.

Answer to Question 01

($\Rightarrow$) We need to prove that: *if n is divisible by a natural number less than or equal to $\sqrt{n}$, then n is composite.* Whenever $\exists a \in \mathbb{N}$ such that

$$\begin{aligned} & a|n \ \wedge \ a \leq \sqrt{n} \\ \implies & a|n \ \wedge \ a < n. \end{aligned}$$

By definition it follows that, n is a composite.

($\Leftarrow$) We need to prove that: *if n is composite, then n is divisible by a natural number less than or equal to $\sqrt{n}$.* Since n is composite, $\exists a \in \mathbb{N},\ 1 < a < n$ such that $a|n$. Here we have either of the following two cases:

$$\begin{aligned} & a^2 = n \ \vee \ a \cdot b = n && \ldots\ldots \text{for some } b \in \mathbb{N} \\ \implies & a = \sqrt{n} \ \vee \ a \cdot a < a \cdot b = n && \ldots\ldots \text{wlog } a < b \\ \implies & a = \sqrt{n} \ \vee \ a^2 < n \\ \implies & a = \sqrt{n} \ \vee \ a < \sqrt{n}. \end{aligned}$$

Hence, a is the required divisor of n which is less than or equal to $\sqrt{n}$.
This completes the proof. *qed*

Question 02 **[5 marks]**

Prove that: For finite sets A, B, there exists a function $f : A \cap B \to A \cup B$ such that f is injective.

Answer to Question 02

Here, we can provide an existential proof. For arbitrary but non-empty sets A, B, we always have $(A \cap B) \subseteq (A \cup B)$. Consider $f : (A \cap B) \to (A \cup B)$ to be the identity function.
For $x_1, x_2 \in A \cap B$ $f(x_1) = f(x_2) \implies x_1 = x_2$, hence this considered f is injective.
This completes the proof. *qed*

Question 03 **[5 marks]**

Let $\{0,1\}^n$ be the set of all binary strings of exactly length n. Define a function $f : \{0,1\}^n \to \mathbb{Z}$ such that:

1. $f(w) \neq 0$ for all $w \in \{0,1\}^n$, and
2. $\sum_{w \in \{0,1\}^n} f(w) = 0$.

Answer to Question 03

The function $f : \{0,1\}^n \to \mathbb{Z}$ can be defined as follows:

$$f(w) = \begin{cases} 1, & \text{if } w \in \{0,1\}^n \text{ starts with } 0, \\ -1, & \text{if } w \in \{0,1\}^n \text{ starts with } 1. \end{cases}$$

Note that there can be other correct answers too!

Question 04 **[5 marks]**

Consider the function $f : \mathbb{Z} \to \mathbb{Z}$ defined as $f(x) = x^2$. Let $A = \{0, 1, 2^2, 3^2, \ldots, n^2\}$ and $B = \{0, 1, 2^2, 3^2, \ldots, n^2, (n+1)^2\}$ for some fixed $n \in \mathbb{N}$. Determine the set $f^{-1}(A) \Delta f^{-1}(B)$.

▶ Notation: Δ denotes the set symmetric difference.

Answer to Question 04

Note that the given function f is not necessarily an injection. Hence we have:

$$f^{-1}(A) = \{-n,\ -n+1, \ldots,\ -1,\ 0,\ 1, \ldots,\ n-1,\ n\}$$
$$f^{-1}(B) = \{-n-1,\ -n,\ -n+1, \ldots,\ -1,\ 0,\ 1, \ldots,\ n-1,\ n,\ n+1\}$$

Note that $f^{-1}(A) \subset f^{-1}(B)$. Hence we have:

$$\begin{aligned} f^{-1}(A) \Delta f^{-1}(B) &= \left(f^{-1}(A) \cup f^{-1}(B)\right) \setminus \left(f^{-1}(A) \cap f^{-1}(B)\right) \\ &= f^{-1}(B) \setminus f^{-1}(A) \\ &= \{-n-1,\ n+1\} \end{aligned}$$

$$\boxed{\therefore\ f^{-1}(A) \Delta f^{-1}(B) = \{-n-1,\ n+1\}}$$

Question 05 **[5 marks + 1 bonus]**

Consider the following function f defined on the set $\{0,1\}^n$ as: for $x \in \{0,1\}^n$, $f(x) = \mathsf{wt}(x)$, where $\mathsf{wt}(x)$ denotes the number of 1's in the string x.

1. Determine the range of f
2. Is f injective ?
3. Determine $f^{-1}(1)$.
4. Determine the size of $f^{-1}(A)$ where $A = \{1, n\}$.

Answer to Question 05

1. There can be $n-$bit binary string which can have no 1s, one 1, two 1s and so on until the case where the string has all n bits to 1. Hence, $Range(f) = \{0, 1, 2, \ldots, n\}$.
2. Clearly, $f(1000\ldots0) = f(0000\ldots1) = 1$. Hence, f is not injective.
3. For $i = 1, 2, 3, \ldots, n$, we define $w_i := \{x \mid x \in \{0,1\}^n \ \wedge \text{only the } i^{th} \text{ bit in } x \text{ is } 1\}$. Observe, that $\forall x \in \{0,1\}^n,\ f(x) = 1$ if and only if $x = w_i$. Hence, $f^{-1}(1) = \{w_i \mid i = 1, 2, 3, \ldots, n\}$.
4. Since $A = \{1\} \cup \{n\}$ and $\{1\}$ and $\{n\}$ are disjoint subsets of $\mathbb{N}$ we have, $f^{-1}(A) = f^{-1}(\{1\}) \cup f^{-1}(\{n\})$. Since, $f^{-1}(\{n\}) = \{111\cdots11\}$ continuing from part(3) we have
$$f^{-1}(A) = \{w_i \mid i = 1, 2, \ldots, n\} \cup \{111\cdots11\}.$$

Question 06 **[5 marks]**

Consider the following recurrence

$$f(n) = \begin{cases} 1 & \text{if } n = 1 \\ m \cdot f(n-1) & \text{if } n \geq 2 \end{cases},$$

where $m \in \mathbb{N}$ is a fixed natural number. Determine a closed form expression for f.

Answer to Question 06

First of all, we perform unrolling to claim a closed form for f.

$$\begin{aligned} f(n) &= m \cdot f(n-1) \\ &= m[m \cdot f(n-2] = m^2 \cdot f(n-2) \\ &= m^3 \cdot f(n-3) \\ &= m^4 \cdot f(n-4) \\ &\vdots \\ &= m^k \cdot f(n-k) \\ &= m^{n-1} \cdot f(1) = m^{n-1} \qquad\qquad [\mathit{substitution}\ n-k=1] \end{aligned}$$

So, after performing unrolling we claim that the closed form of f is $f(n) = m^{n-1}$. Now we use induction to prove our claim.

BASE CASE : $m^{1-1} = 1 = f(1)$. Hence, the claim is valid for the base case.

IH : We assume that $\exists k \in \mathbb{N}$ such that $f(k) = m^{k-1}$.

Inductive Step : $f(k+1) = m \cdot f(k) = m\big(\underbrace{m^{k-1}}_{IH}\big) = m^{(k+1)-1}$.

This completes the proof and the determined closed form is $f(n) = m^{n-1}$.

Question 07 **[5 marks]**

Determine the number of functions

$$f : \{1, 2, 3, 4, 5, 6\} \to \{1, 2, 3, 4, 5, 6\}$$

such that $f(1) = 2$.

Answer to Question 07

For each $n = 2, 3, 4, 5, 6$ observe that $f(n)$ has 6 mapping possibilities to the elements in the given codomain. Hence, the required number of plausible functions is 6^5.

OR

Question 08 **[2 + 2 + 1 = 5 marks]**

The Set Symmetric Difference Δ is defined as follows:

$$A \Delta B = (A \cup B) \setminus (A \cap B).$$

1. What can be said about sets A and B if $A \Delta B = A$?
2. What can be said about sets A and B if $A \Delta B = A \setminus B$?
3. For any non-empty set A, what is $A \Delta A$?

▶ Notation: $A \setminus B$ denotes A set minus B.

Answer to Question 08

1. $\mathbf{B} = \phi$.
   We have $A \Delta B = A$. It follows that

$$\begin{aligned}
& (A \cup B) \setminus (A \cap B) && = A \\
\implies & (A \cup B) \cap (A \cap B)^c && = A \\
\implies & (A \cup B) \cap (A^c \cup B^c) && = A \quad \ldots (De\ Morgan's\ Law) \\
\implies & \big((A \cup B) \cap A^c\big) \cup \big((A \cup B) \cap B^c\big) && = A \quad \ldots (Distributive\ Law) \\
\implies & \underbrace{(B \cap A^c)}_{I} \cup \underbrace{(A \cap B^c)}_{II} && = A \quad \ldots (on\ simplifying)
\end{aligned}$$

   Since, $I \subseteq A^c$ and $II \subseteq A$ we have:

$$\begin{aligned}
& I = \phi \ \land \ II = A \\
\implies & B = \phi.
\end{aligned}$$

2. $\mathbf{B} \subseteq \mathbf{A}$.
   We have $A \Delta B = A$. It follows that

$$\begin{aligned}
& (A \cup B) \setminus (A \cap B) && = A \setminus B \\
\implies & (A \cup B) \cap (A \cap B)^c && = A \setminus B \\
\implies & (A \cup B) \cap (A^c \cup B^c) && = A \setminus B \quad \ldots (De\ Morgan's\ Law) \\
\implies & \big((A \cup B) \cap A^c\big) \cup \big((A \cup B) \cap B^c\big) && = A \setminus B \quad \ldots (Distributive\ Law) \\
\implies & \underbrace{(B \cap A^c)}_{I} \cup \underbrace{(A \cap B^c)}_{II} && = A \setminus B \quad \ldots (on\ simplifying)
\end{aligned}$$

   Since, $I \subseteq A^c$ and by definition $II = A \setminus B$ we have:

$$\begin{aligned}
& I = \phi \\
\implies & A^c \cap B = \phi \\
\implies & B \subseteq A.
\end{aligned}$$

3. $\boldsymbol{\phi}$.

$$\begin{aligned}
A \Delta A &= (A \cup A) \setminus (A \cap A) \\
&= A \setminus A \\
&= A \cap A^c \\
&= \phi.
\end{aligned}$$

7.7 Quiz 03

Question 01 **[5 + 3 + 4 + 3 = 15 marks]**

1. Show that if $f(n) = n^2 \log_2 n - 5n^2 + 3n \log_2 n$, then for $c_1 = 1/2$ and $c_2 = 3$, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, we have

$$c_1 \cdot n^2 \log_2 n \leq f(n) \leq c_2 \cdot n^2 \log_2 n.$$

Determine the smallest computable value of n_0, so that the above inequalities establish $f(n)$ is $\Theta(n^2 \log_2 n)$.

First Inequality: $c_1 \cdot n^2 \log_2 n \leq f(n)$.
It suffices to show that,
$\frac{1}{2} n^2 \log_2 n \leq n^2 \log_2 n - 5n^2$,
implies that $1/2 \leq 1 - \frac{5}{\log_2 n}$.
Solving further we get $2^{10} \leq n$. Thus $n_1 = 1024$.

Second Inequality: $f(n) \leq c_2 \cdot n^2 \log_2 n$.
$\forall n \geq 1$, we have $n^2 \log_2 n \leq n^2 \log_2 n$; $-5n^2 \leq n^2 \log_2 n$ and $3n \log_2 n \leq n^2 \log_2 n$.
Adding these we get, $f(n) \leq 3 \cdot n^2 \log_2 n$, where $n_2 = 1$

Considering both calculations above, we get $n_0 = max\{n_1, n_2\} = 1024$.

2. If $f(n) = \log_2(n!) + n^2 \log_2 n$, then determine whether $f(n)$ is $O(n \log_2 n)$. Provide a clear justification.

[Hint: If $f_1 \in O(g)$ and $f_2 \in O(g)$, then $f_1 + f_2 \in O(g)$.]
Forall $n \in \mathbb{N}$ we have $n! \leq n^n \implies \log_2(n!) \leq n \log_2 n$. Thus, we have the result $\log_2(n!) \in O(n \log_2 n)$ for $c = 1; n_0 = 1$. Further note that for all $n \in \mathbb{N}$, we have $n \log_2 n < n^2 \log_2 n$, leading to the fact that $n^2 \log_2 n$ can never be $O(n \log_2 n)$.
Proof by contradiction : Suppose $\log(n!) + n^2 \log n = O(n \log n)$.
Hence $n^2 \log n = O(n \log n) - \log(n!)$. As $\log(n!) \in O(n \log n)$, it follows that $n^2 \log n$ is $O(n \log n)$, which is a contradiction.

3. Show that if $f(n) = \log_2 n$ and $g(n) = n^{1/4}$, then $f(n)$ cannot be $\Omega\big(g(n)\big)$.
We claim that $\log_2 n \in o(n^{1/4})$. Consider

$$\begin{aligned}
\lim_{n\to\infty} \frac{\log_2 n}{n^{1/4}} &= \lim_{n\to\infty} \frac{k \cdot 1/n}{1/n^{3/4}} && \text{[L'Hôpital's rule]} \\
&= \lim_{n\to\infty} \frac{4k}{n^{1/4}} = 0. \\
\therefore\ \log_2 n \in o(n^{1/4}) & && \ldots\text{claim proved.}
\end{aligned}$$

By first principle, $\log_2 n \in o(n^{1/4})$ implies that

$$\begin{aligned}
&\forall c > 0,\ \exists\ n_0 > 0\ s.t.\ \forall n > n_0\ |\log_2 n| \leq c \cdot |n^{1/4}| \\
\implies &\forall c > 0,\ \exists\ n_0 > 0\ s.t.\ \forall n > n_0\ \log_2 n \leq c \cdot n^{1/4} \qquad (1)
\end{aligned}$$

Proof by contradiction : Let us assume that $\log_2 n \in \Omega(n^{1/4})$. It follows that $\big(\exists c > 0,\ \exists n_0 > 0\ s.t.\ \forall n > n_0\ \log_2 n \geq c \cdot n^{1/4}\big)$. But this is a clear contradiction to inference (1) derived above. So our assumption is incorrect and hence $\log_2 n$ can never be $\Omega\big(n^{1/4}\big)$. *qed*

4. Show that if $f(n) = n^{\frac{3}{\log_2 n}}$, then $f(n)$ is $O(1)$.
$f(n) = n^{\frac{3}{\log_2 n}} = n^{3 \cdot \log_n 2} = 2^3$. Clearly, for $\forall n \geq n_0$ where $n_0 = 1$ and $c = 10$ we have $f(n) \leq c \cdot 1$. Hence, $f(n) \in O(1)$. *qed*

OR

Question 02 **[5 + 5 + 5 = 15 marks]**

For some fixed $n \in \mathbb{N}$, $p(x) : \mathbb{R} \to \mathbb{R}$ defined as $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ is called a real-valued **n-degree polynomial**, where $a_i \in \mathbb{R}$ for all $i = 1, 2, \ldots, n$.

For the following three questions consider the cost of binary multiplication `a * b` to be $O(1)$, and ignore any complexity due to any binary addition, variable increment and variable initiation.

1. The conventional algorithm for evaluating a polynomial $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ at $x = c$ can be expressed in the following pseudocode (Algorithm-I) :

```
Algorithm-I
1:  procedure EVAL.POLYNOMIAL(c, a_0, a_1, ..., a_n: reals)
2:      y := 0                                          ▷ initiation
3:      for i := 1 to n do
4:          power := 1                                  ▷ initiation
5:          for j := 1 to i do
6:              power = power * c
7:          end for
8:          y := y + a_i * power
9:      end for
10:     return y          ▷ y = a_n c^n + a_{n-1} c^{n-1} + ··· + a_1 c + a_0
11: end procedure
```

   (a) What is the total number of multiplications and additions done by this algorithm to evaluate a n-degree polynomial $p(x)$ at a given point $x = c$?
   The multiplication at `line 6` is happening once for each iteration of the loop at `line 5`. Further for each iteration of the loop at `line 3`, the loop at `line 5` is restarting and the multiplication at `line 8` is happening once.
   The addition at `line 8` is happening one for each iteration of the loop at `line 3`. Hence we have,

$$\text{total no. of multiplications} = (1+1) + (2+1) + (3+1) + \cdots + (n+1)$$
$$= \sum_{i=1}^{n} i + n = \frac{n^2 + 3n}{2},$$
$$\text{total no. of additions} = n.$$

   (b) What is the complexity of this conventional algorithm (Algorithm-I) in Θ-bound?
   Since binary multiplication is the only cost inducing operation, observe that $\forall n \geq 1$ we have

$$\frac{1}{10}n^2 \leq \frac{1}{2}n^2 + \frac{3}{2}n \leq 2n^2.$$

   Hence, complexity of Algorithm-I is $\Theta\big(n^2\big)$.

2. The **Horner's method** of evaluating a polynomial $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ at $x = c$ can be expressed in the following pseudocode (Algorithm-II) :

Algorithm-II

1: **procedure** HORNER($c, a_0, a_1, \ldots, a_n$: reals)
2: $\quad y := a_n$ ▷ initiation
3: $\quad$ **for** $i := 1$ to n **do**
4: $\quad\quad y = y * c + a_{n-i}$
5: $\quad$ **end for**
6: $\quad$ **return** y ▷ $y = a_n c^n + a_{n-1} c^{n-1} + \cdots + a_1 c + a_0$
7: **end procedure**

(a) What is the total number of multiplications and additions done by this algorithm to evaluate a n-degree polynomial $p(x)$ at a given point $x = c$?
Both the multiplication and the addition at `line 4` is happening once for each iteration of the loop at `line 3`. Hence we have,

$$\text{total no. of multiplications} = n,$$
$$\text{total no. of additions} = n.$$

(b) What is the complexity of this Horner's algorithm (Algorithm-II) in Θ-bound?
Since binary multiplication is the only cost inducing operation, observe that $\forall n \geq 1$ we have

$$\frac{1}{2}n \leq n \leq 2n.$$

Hence, complexity of Algorithm-II is $\Theta\big(n\big)$.

3. An optimised algorithm for evaluating a polynomial $p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ at $x = c$ can be expressed in the following pseudocode (Algorithm-III) :

Algorithm-III

1: **procedure** EVAL.POLYNOMIAL($c, a_0, a_1, \ldots, a_n$: reals)
2: $\quad power := 1$ ▷ initiation
3: $\quad y := a_0$ ▷ initiation
4: $\quad$ **for** $i := 1$ to n **do**
5: $\quad\quad power = power * c$
6: $\quad\quad y = y + a_1 * power$
7: $\quad$ **end for**
8: $\quad$ **return** y ▷ $y = a_n c^n + a_{n-1} c^{n-1} + \cdots + a_1 c + a_0$
9: **end procedure**

(a) What is the total number of multiplications and additions done by this algorithm to evaluate a n-degree polynomial $p(x)$ at a given point $x = c$?
Observe that for each iteration of the loop at `line 4`, we have two multiplications (one each from `line 5` and `line 6`) and one addition (`line 6`). Hence we have,

$$\text{total no. of multiplications} = 2n,$$
$$\text{total no. of additions} = n.$$

(b) What is the complexity of this optimised algorithm (Algorithm-III) in Θ-bound?
Since binary multiplication is the only cost inducing operation, observe that $\forall n \geq 1$ we have

$$n \leq 2n \leq 3n.$$

Hence, complexity of Algorithm-III is $\Theta(n)$.

Question 03 **[10 marks]**

Consider the problem $\mathsf{Select}_{n,i}$ described as follows:

$\mathsf{Select}_{n,i}$
Input: An n-size array B of distinct integers; An i in $1 \leq i \leq n$
Output: The ith smallest element in B

Suppose we have an algorithm $\mathcal{A}$ that finds the smallest element in a given array of integers, i.e., $\mathcal{A}$ is an algorithm for solving $\mathsf{Select}_{n,1}$. Using $\mathcal{A}$, we can design an algorithm $\mathcal{A}'$ to solve $\mathsf{Select}_{n,i}$ as follows:

```
Algorithm A'
1: for j := 1 to i do
2:     Find a_j by running A on B
3:     Remove a_j from B
4: end for
5: return a_i
```

Show that if $f(n) = n - 1$ is the computational complexity of $\mathcal{A}$ for solving $\mathsf{Select}_{n,1}$, then computational complexity of $\mathcal{A}'$ is $\frac{2ni-i^2-i}{2}$.
The removal operation at `line 3` has no cost, but it reduces the size of array B by 1 in each iteration of the loop at `line 1`. The mechanics of the operation at `line 1` in the loop at `line 1` is as follows:

Iteration no.	Size of problem at `line 2`	Cost, i.e., $f(size)$
1	n	$n-1$
2	$n-1$	$n-2$
3	$n-2$	$n-3$
$\vdots$	$\vdots$	$\vdots$
i	$n-i+1$	$n-i$

Thus the total cost of $\mathcal{A}'$ gives its complexity to be

$$(n-1) + (n-2) + \cdots + (n-i) = \frac{2ni - i^2 - i}{2}.$$

Question 04 **[10 marks]**

Draw a recursion tree table for the following function, in the format given below.

$$f(n) = \begin{cases} 8f(n/2) + n^3 & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

Use the table to show that $f(n) = \Theta(n^3 \log n)$. Assume n is always a power of 2.

Format for Recursion Tree Table

Level	No. of subproblems	Size of subproblems	Cost per subproblem	Cost per level

Observe that the given recurence unrolls as follows:

$$\begin{aligned} f(n) &= 8 \cdot f(n/2) + n^3 \\ 8 \cdot f(n/2) &= 64 \cdot f(n/4) + 8 \cdot (n/2)^3 \\ 64 \cdot f(n/4) &= 512 \cdot f(n/8) + 64 \cdot (n/4)^3 \\ &\vdots \qquad\qquad \vdots \\ 8^k \cdot f(n/2^k) &= 8^{k+1} \cdot f(n/2^{k+1}) + 8^k \cdot (n/2^k)^3. \end{aligned}$$

For the given recurence we the following at level k:

$$\boxed{8^k \cdot f(\underbrace{n/2^k}_{\text{II}}) = 8^{k+1} \cdot f(n/2^{k+1}) + \overbrace{\underbrace{8^k}_{\text{I}} \cdot \underbrace{(n/2^k)^3}_{III}}^{IV}.}$$

I : no. of subproblems
II : size of subproblems
III : cost per subproblems
IV : cost per level = no. of subproblems $\times$ cost per subproblems

Level	No. of subproblems	Size of subproblems	Cost per subproblem	Cost per level
0	1	n	n^3	$1 \cdot n^3$
1	8	$\frac{n}{2}$	$\left(\frac{n}{2}\right)^3$	$8 \cdot \left(\frac{n}{2}\right)^3 = n^3$
2	8^2	$\frac{n}{4}$	$\left(\frac{n}{4}\right)^3$	$8^2 \cdot \left(\frac{n}{4}\right)^3 = n^3$
3	8^3	$\frac{n}{8}$	$\left(\frac{n}{8}\right)^3$	$8^3 \cdot \left(\frac{n}{8}\right)^3 = n^3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\log_2 n$	$8^{\log_2 n} = n^3$	$\frac{n}{2^{\log_2 n}} = 1$	$\left(\frac{n}{2^{\log_2 n}}\right)^3 = 1$	$8^{\log_2 n} \cdot \left(\frac{n}{2^{\log_2 n}}\right)^3 = n^3$

The required recursion tree table.

Using the table above, we get the total cost of the recurence

$$\begin{aligned} &= n^3 + 8 \cdot \left(\frac{n}{2}\right)^3 + 8^2 \cdot \left(\frac{n}{4}\right)^3 + \cdots + 8^{\log_2 n} \cdot \left(\frac{n}{2^{\log_2 n}}\right)^3 \\ &= n^3 \cdot \left[1 + 8 \cdot \left(\frac{1}{2}\right)^3 + 8^2 \cdot \left(\frac{1}{4}\right)^3 + \cdots + 8^{\log_2 n} \cdot \left(\frac{1}{2^{\log_2 n}}\right)^3\right] \\ &= n^3 \cdot [\underbrace{1 + 1 + 1 + \cdots + 1}_{\log_2 n \ + \ 1 \text{ times}}] \\ &= n^3 \cdot \log_2 n \ + \ n^3 \\ &= \Theta(n^3 \log_2 n) \qquad\qquad \ldots\ldots [c_1 = 1, c_2 = 5; n_0 = 3] \end{aligned}$$

7.8 Quiz 04

Question 01 **[10 marks]**

Let $\Omega = \{0,1\}^4$ be the set of all binary strings of length 4. Consider the *experiment of generating binary strings of length* 4 which realises the uniform probability distribution. Let $A \subseteq \Omega$ be the event that the generated binary string starts with 1 and $B \subseteq \Omega$ be the event that the generated binary string contains even number of 1s. Under the assumption that appearances of 0s and 1s are independant and uniform, determine whether A and B are independant or not.

► Note: The string 0000 is considered as having even number of 1-bits.

Answer: yes.

$|\Omega| = 2^4 = 16$;

$A = \{1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}$ and $\mathbb{P}[A] = 1/2$;

$B = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\}$ and $\mathbb{P}[B] = 1/2$;

$A \cap B = \{1001, 1010, 1100, 1111\}$ and $\mathbb{P}[A \cap B] = 1/4$.

Since $\mathbb{P}[A] \cdot \mathbb{P}[B] = \mathbb{P}[A \cap B]$, we have events A and B to be independant.

OR

Question 02 **[5 + 5 = 10 marks]**

1. It is given that A, B, C are pairwise independent, with $\mathbb{P}[A] = \mathbb{P}[B] = \mathbb{P}[C] = 1/2$. Further if events A and $B \cup C$ are also independent, then compute $\mathbb{P}[A \cap (B \cup C)]$.
   Answer : 3/8.

$$\begin{aligned}
\mathbb{P}[A \cap (B \cup C)] &= \mathbb{P}[A] \cdot \mathbb{P}[(B \cup C)] && \text{[due to independence]} \\
&= \frac{1}{2} \cdot \Big[\mathbb{P}[A] + \mathbb{P}[B] - \mathbb{P}[(B \cap C)\Big] && \text{[Inclusion-Exclusion principle]} \\
&= \frac{1}{2} \cdot \Big[\mathbb{P}[A] + \mathbb{P}[B] - \mathbb{P}[B] \cdot \mathbb{P}[C]\Big] && \text{[due to independence]} \\
&= \frac{1}{2} \cdot \left[\frac{1}{2} + \frac{1}{2} - \frac{1}{2} \cdot \frac{1}{2}\right] \\
&= 3/8.
\end{aligned}$$

2. Random variable X takes values in the set $\{-3, -2, 1, 2, 3\}$ with $\mathbb{P}_X[2] = \mathbb{P}_X[-2]$ and $\mathbb{P}_X[3] = \mathbb{P}_X[-3]$. If $E[X] = 1/4$, then compute $\mathbb{P}_X[1]$.
   Answer : 1/4.

$$\begin{aligned}
E(X) &= -3 \cdot \mathbb{P}_X[-3] - 2 \cdot \mathbb{P}_X[-2] + \mathbb{P}_X[1] + 2 \cdot \mathbb{P}_X[2] + 3 \cdot \mathbb{P}_X[3] \\
&= \underbrace{-3 \cdot \mathbb{P}_X[3] - 2 \cdot \mathbb{P}_X[2] + 2 \cdot \mathbb{P}_X[2] + 3 \cdot \mathbb{P}_X[3]}_{\textit{equals } 0} + \mathbb{P}_X[1] \\
&= \mathbb{P}_X[1].
\end{aligned}$$

   It is given that $E(X) = 1/4$, hence $\mathbb{P}_X[1] = 1/4$.

Question 03 **[7 + 3 = 10 marks]**

1. For a fixed $n \in \mathbb{N}$, let X be a random variable such that $Range(X) := \{1, 2, 3, \cdots, n\}$ and $\mathbb{P}_X$ is the Uniform probability distribution. Show that $\sum_{i=1}^{n} \mathbb{P}_X[X \geq i] = \frac{n+1}{2}$.
Since probability distributes over disjoint events, we begin as follows:

$$
\begin{aligned}
\sum_{i=1}^{n} \mathbb{P}_X[X \geq i] &= \mathbb{P}_X[X \geq 1] + \mathbb{P}_X[X \geq 2] + \cdots + \mathbb{P}_X[X \geq n] \\
&= \left(\sum_{i=1}^{n} \mathbb{P}_X[i]\right) + \left(\sum_{i=2}^{n} \mathbb{P}_X[i]\right) + \cdots + \left(\sum_{i=n}^{n} \mathbb{P}_X[i]\right) \\
&= 1 \cdot \mathbb{P}_X[1] + 2 \cdot \mathbb{P}_X[2] + \cdots + n \cdot \mathbb{P}_X[n] \\
&= 1 \cdot \left(\frac{1}{n}\right) + 2 \cdot \left(\frac{1}{n}\right) + \cdots + n \cdot \left(\frac{1}{n}\right) \qquad [\because \mathbb{P}_X \sim U] \\
&= \frac{1}{n} \cdot \frac{n(n+1)}{2} \\
&= \frac{n+1}{2}.
\end{aligned}
$$

2. Consider $\mathbb{P} : \Omega \to [0, 1]$ to be a probability distribution on a non-empty sample space Ω. If $A, B, C \subseteq \Omega$ be events such that $A \cap \bar{C} = B \cap \bar{C}$, then show that $|\mathbb{P}(A) - \mathbb{P}(B)| \leq \mathbb{P}(C)$.
Note that, $|\mathbb{P}(A) - \mathbb{P}(B)| \leq \mathbb{P}(C) \Leftrightarrow -\mathbb{P}[C] \leq \mathbb{P}[A] - \mathbb{P}[B] \leq \mathbb{P}[C]$. Hence, consider both of the following calculations.

$$
\begin{aligned}
&A \cap \bar{C} = B \cap \bar{C} \\
\Rightarrow &\mathbb{P}[A \cap \bar{C}] = \mathbb{P}[B \cap \bar{C}] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[A \cap C] = \mathbb{P}[B] - \mathbb{P}[B \cap C] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[B] = \mathbb{P}[A \cap C] - \mathbb{P}[B \cap C] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[B] \leq \mathbb{P}[A \cap C] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[B] \leq \mathbb{P}[A \cap C] \leq \mathbb{P}[C]
\end{aligned}
$$

$$
\begin{aligned}
&A \cap \bar{C} = B \cap \bar{C} \\
\Rightarrow &\mathbb{P}[A \cap \bar{C}] = \mathbb{P}[B \cap \bar{C}] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[A \cap C] = \mathbb{P}[B] - \mathbb{P}[B \cap C] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[B] = \mathbb{P}[A \cap C] - \mathbb{P}[B \cap C] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[B] \geq -\mathbb{P}[A \cap C] \\
\Rightarrow &\mathbb{P}[A] - \mathbb{P}[B] \geq -\mathbb{P}[A \cap C] \geq -\mathbb{P}[C]
\end{aligned}
$$

This completes the proof. *qed.*

Question 04 **[15 marks]**

An experiment Π has sample space Ω such that $|\Omega| = m$, where m is a fixed natural number. Let X and Y be random variables on Ω such that $Range(X) := \{1, 2, 3, \cdots, m\}$ and $Range(Y) := \{\mathbb{P}_X(1), \mathbb{P}_X(2), \mathbb{P}_X(3), \cdots, \mathbb{P}_X(m)\}$, where $P_X(i)$, $\forall i = 1, 2, 3 \cdots, m$ are all distinct. Show that the expected value, $E(Y) = 1/m$ when it is given that $\mathbb{P}_Y$ is the Uniform probability distribution.

$$\begin{tikzcd}
\{\mathbb{P}_X(1), \mathbb{P}_X(2), \dots, \mathbb{P}_X(m)\} \arrow[d, "\mathbb{P}_Y \sim U"] & \underset{\{\omega_1, \omega_2, \dots, \omega_m\}}{\Omega} \arrow[l, "Y"'] \arrow[r, "X"] \arrow[d, "P"] & \{1, 2, 3, \dots, m\} \arrow[d, "\mathbb{P}_X"] \\
{[0,1]} & {[0,1]} & {[0,1]}
\end{tikzcd}$$

The given scenario can be represented using the block diagram given above. Thus, we consider the following:

$$\begin{aligned}
E(Y) &= \sum_{i=1}^{m} Y(\omega_i) \cdot \mathbb{P}_Y[Y(\omega_i)] && \dots [\text{by definition}] \\
&= \sum_{i=1}^{m} \left(\mathbb{P}_X(i) \cdot \frac{1}{m} \right) && \dots [\mathbb{P}_Y \sim U] \\
&= \frac{1}{m} \cdot \underbrace{\left(\sum_{i=1}^{m} \mathbb{P}_X(i) \right)}_{=1} && \dots [\mathbb{P}_X \text{ is a prob. dist.}] \\
&= \frac{1}{m}.
\end{aligned}$$

PROGRAMING PROBLEM SETS

8.1 Problem Set 01 - Intro to loops & conditional statements

1. Print the factors of a positive integer given as input.
2. Print the sum of digits in an input positive integer.
3. Count the number of digits in an integer given as input.
4. Check if a input number is a Perfect Number.
5. Check if an input positive integer is an Armstrong Number.
6. Reverse the digits of a positive integer given as input.

8.2 Problem Set 02 - Nested loops-I

1. Write a program in C programming language to evaluate the logical expression
$$(A \wedge B) \vee \neg C$$
for given boolean values of A, B and C as inputs. Your code must take three integers (either 0 or 1) as inputs from the user and shall return the truth value of the given logical expression as result.
2. Write a program in C programming language to check whether a given number is divisible by 5, 13, both or neither. Your code must take an integers as input from the user and shall return the status of divisibility mentioned as result.
3. Write a program in C programming language which prints the complete truth table of the following logical expressions:
   (a) $(A \vee B) \wedge \neg C$
   (b) $(A \wedge \neg B) \rightarrow (B \vee \neg(A \wedge B))$
   (c) $(A \wedge B) \vee \neg C$
4. By now each one of you must be familiar about semantic entailment and how it can be used to prove any particular logical argument to be invalid. Write a program in C programming language, which uses semantic entailment to prove invalidity of the following logical arguments.
   (a) $(r \rightarrow (\neg p \vee q)),\ (p \wedge \neg r) \models \neg(\neg r \vee q)$
   (b) $(p \vee (q \wedge \neg r)),\ (r \rightarrow (\neg q \vee p)) \models (q \wedge \neg p)$
   (c) $(\neg p \vee q),\ (\neg r \vee \neg q),\ (\neg r \vee p) \models \neg r$

► 0 is the boolean value for **False** in C.

Logical Operators	Symbol in theory	Symbol in C syntax
neg	$\neg$	!
conjunction	$\wedge$	&&
disjunction	$\vee$	||

8.3 Problem Set 03 - Nested loops-II

1. Write a program in C programming language to check if a positive integer given as input is a Palindrome or not. Your code must take a positive integer as input from the user and shall print as output if it is a palindrome number or not.

2. Write a program in C programming language to check if a positive integer given as input is a Prime or not. Your code must take a positive integer as input from the user and shall print as output if it is a prime number or not. 1 is not a prime, by definition.

3. Write a program in C programming langauge to print the Fibonacci Series up to N Terms. Your code must take a positive integer as input from the user - **N** and shall print as output the Fibonacci Series as per the mentioned requirement.

   (a) Try to code the above **without** using arrays.

   (b) Try to code the above using 1-Dimensional arrays.

4. Write a program in C programming language to print all possible 4-bit binary strings, that is, binary strings of length 4. **Hint**: at best it only requires nested for loops.

8.4 Problem Set 04 - Intro to 1D arrays & more on nested loops

1. Write a program in C programming language which takes two binary strings as inputs from the user and performs the following bitwise binary operations and gives the result as output. The concept of loops, nested loops and 1D arrays shall suffice.

   (a) Bitwise AND

   (b) Bitwise OR

   (c) Bitwise XOR

   (d) Bitwise NAND

   (e) Bitwise NOR

   (f) Bitwise XNOR

2. Write a program in C, which approximates the value of $\sin(\pi/2)$ using the following Taylor Series for sine function. Do the approximation with a 0.01% precision involving the least number of Taylor series terms and print the approximated value, the actual value and the error in approximation. Do not create any implicit custom functions in your code. ⋆

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} \cdot x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \cdots$$

   - Use the header file `math.h`, which has in-built function `sin()` to get the actual value of $\sin(\pi/2)$.
   - The function `fabs()` from `math.h` is used for modulus or the absolute value of its argument.
   - The constant π is stored as `M_PI` in `math.h`.
   - error in approximation = |approximated value - actual value| .

3. Write a program in C, which approximates the value of $\cos(\pi/2)$ using the following Taylor Series for cosine function. Do the approximation with a 0.01% precision involving the least number of Taylor series terms and print the approximated value, the actual value and the error in approximation. Do not create any implicit custom functions in your code. $\star$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} \cdot x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \cdots$$

   - Use the header file `math.h`, which has in-built function `cos()` to get the actual value of $\cos(\pi/2)$.
   - The function `fabs()` from `math.h` is used for modulus or the absolute value of its argument.
   - The constant π is stored as `M_PI` in `math.h`.
   - error in approximation = |approximated value - actual value| .

4. Write a program in C, which takes members of two sets as inputs and stores them in two 1D-arrays, and gives all the members of their cross-product as output. Cardinality of each input set is 4.

5. Write a program in C, which takes members of a set as inputs and stores them in a 1D-array, and gives all the members of its power set as output. Cardinality of the input set is 3. $\star\star$

Logical Operators	Symbol in theory	Symbol in C syntax
Bitwise NOT	$\neg$	~
Bitwise AND	$\wedge$	&
Bitwise OR	$\vee$	|
Bitwise XOR	$\oplus$	^

8.5 Problem Set 05 - Nested loops and theory related computation problems

1. For positive real numbers a and r (where $0 < r < 1$), the sum of the infinite geometric series is given by:

$$a + ar + ar^2 + \cdots = \frac{a}{1-r}.$$

   Write a program in C that takes three real numbers a, r, c as inputs, where $a > 0$, $0 < r < 1$, and $0 < c < 1$. The program should output the smallest integer k such that

$$\left(\frac{a}{1-r}\right) - (a + ar + ar^2 + \cdots + ar^k) < c.$$

   Additionally, the program must print the values of $(\frac{a}{1-r})$ and the partial sum $(a+ar+ar^2+\cdots+ar^k)$. $\star$

2. Write a program in C programming language which takes two positive integers k and n as inputs from the user, and outputs the equivalence class of $\mathbb{Z}_n$ in which k belongs. For example, for $n = 7$, $\mathbb{Z}_7 = \{\hat{0}, \hat{1}, \hat{2}, \hat{3}, \hat{4}, \hat{5}, \hat{6}\}$ has 7 equivalence classes and $k = 25 \in \hat{4}$. Do not use the in-built '%' operator, even once in the entire program.

3. For $x \in \mathbb{R}$, the Taylor Series for exponential function is given by:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

   Write a program in C that takes c $(0 < c < 1)$ and $x \in \mathbb{R}^+$ as inputs and computes the above series for until the following happens for the smallest possible integer k:

$$\Big| \underbrace{(e^x)}_{actual} - \underbrace{(1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^k}{k!})}_{approx.} \Big| < c.$$

   As output, print the actual and the approximated value for e^x. Use the header file `math.h`, which has in-built function `exp()` to get the actual value of e^x and the function `fabs()` for the absolute value of its argument. Both needs argument of double data type.

4. Write a program in C programming language which takes a binary string of length 5 as input, stores it in a 1D array and checks whether it is palindrome or not. *Can this be extended such that the program takes a positive integer as input, and checks if it is a palindrome or not?* $\star$

8.6 Problem Set 06 - Introduction to functions in C

Definition 8.6.1. *($\mathcal{O}-notation$) For functions $f : \mathbb{N} \to \mathbb{R}$ and $g : \mathbb{N} \to \mathbb{R}$ we say that*

$$f(x) = \mathcal{O}\big(g(x)\big),$$

if there exists a constant $C \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that we have $|f(n)| \leq C \cdot |g(n)|$, $\forall n \geq n_0$.

Lemma 8.6.2. *All polynomials are $\mathcal{O}$ of their term with the highest power. Let $f : \mathbb{N} \to \mathbb{R}$ be a polynomial of degree k, i.e, $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_kx^k$, where $a_i's$ are real numbers, and $g : \mathbb{N} \to \mathbb{R}$ be the $k-$degree monomial, i.e, $g(x) = x^k$, then $\forall k \in \mathbb{N}$ we have*

$$f(x) = \mathcal{O}\big(g(x)\big).$$

Proof. *left as homework.*

1. Write a program in C programming language which:

   step i. takes positive integer k as input,
   step ii. takes the real constants $a_0, a_1, \cdots, a_k$ as inputs and stores them in an 1D array,
   step iii. takes a real number $r > |a_k| + k$ as input and then finally

   gives the smallest natural number n_0 as output for which $a_0 + a_1x + a_2x^2 + \cdots + a_kx^k$ is $\mathcal{O}(x^k)$.

2. The following given are the recursive and the closed forms of some functions from $\mathbb{N}$ to $\mathbb{N}$. Write programs in C programming language for each of them which takes a positive integer k as input and computes the image of the function using their recursive forms and closed forms. Define and use custom made implicit functions in your program appropriately.

.	Function	Recursive Form	Closed Form	Comment
1.	$S(n)$	$S(n-1)+n$	$\frac{n(n+1)}{2}$	sum of first n natural no.s
2.	$S_0(n)$	$S_0(n-1)+(2n-1)$	n^2	sum of first n odd natural no.s
3.	$S_s(n)$	$S_s(n-1)+n^2$	$\frac{n(n+1)(2n+1)}{6}$	sum of squares of first n natural no.s

3. Write a program in C programming language which takes a positive integer n as input and gives as output the n^{th} term of the Fibonacci sequence, computed both recusively and iteratively. Define and use custom made implicit functions in your program for each of the two ways of computation. **Note:** for Fibonacci sequence $F_1 = F_2 = 1$.

4. Write a program in C programming language which takes a positive integer n as input and gives as output the $n!$, computed both recusively and iteratively. Define and use custom made implicit functions in your program for each of the two ways of computation. **Note:** for factorials, $0! = 1$.

8.7 Problem Set 07 - Recursive functions in C-I

1. **Comparisons in Binary Search**
   Let $B(n)$ denote the number of comparisons needed to search an element in a sorted array of size n. The recurence relation $\forall n \in \mathbb{N}$ of this function is:

$$B(n) = \begin{cases} 1 & n = 1, \\ B(n/2) + 1 & \textit{otherwise}. \end{cases}$$

   Also, the closed form for the same function is:

$$B(n) = \log_2 n + 1, \qquad \forall n \in \mathbb{N}.$$

   Write a program in C programming language which takes a positive integer k as input from the user and computes $B(k)$ both using the recursive relation and the closed form and prints them as output.
   ▶ *specifications & notes* :

   1. Define two custom functions, one each for the recurence computation and closed form evaluation.
   2. Use `log2(` _*double*_ `)` from `math.h` for calculating $\log_2 n()$.

2. **Babylonian Method** ⋆
   A square root approximation for any positive rational number r has the following recurence relation:

$$a_i = \begin{cases} r/2 & i = 0, \\ \frac{1}{2}\left(a_{i-1} + \frac{r}{a_{i-1}}\right) & \textit{otherwise}. \end{cases}$$

   Write a program in C programming language which takes a positive reals r and c ($0 < c < 1$) as inputs from the user and recursively approximates $\sqrt{r}$ untill we have

$$\left|\sqrt{r} - a_k\right| \leq c.$$

As output we need the approximated square root and the smallest integer k for which the above happens.

▶ *specifications & notes* :

1. Define custom function for recursive computation.
2. Use `fabs(` *_double_* `)` from `math.h` for absolute value.
3. Use `sqrt(` *_double_* `)` from `math.h` for square root.

3. **Moves in Tower of Hanoi**

Let $H(n)$ denote the number of moves needed to solve the Tower of Hanoi problem with n disks. The recurence relation $\forall n \in \mathbb{N}$ of this function is:

$$H(n) = \begin{cases} 1 & n = 1, \\ 2H(n-1) + 1 & otherwise. \end{cases}$$

Also, the closed form for the same function is:

$$B(n) = 2^n - 1, \qquad \forall n \in \mathbb{N}.$$

Write a program in C programming language which takes a positive integer k as input from the user and computes $H(k)$ both using the recursive relation and the closed form and prints them as output.

▶ *specifications & notes* :

1. Define two custom functions, one each for the recurence computation and closed form evaluation.

4. **Newton-Raphson Method** ⋆

A cube root approximation for any positive rational number r has the following recurence relation:

$$a_i = \begin{cases} r/2 & i = 0, \\ \frac{1}{3}\left(2a_{i-1} + \frac{r}{(a_{i-1})^2}\right) & otherwise. \end{cases}$$

Write a program in C programming language which takes a positive reals r and c ($0 < c < 1$) as inputs from the user and recursively approximates $\sqrt[3]{r}$ untill we have

$$\left|\sqrt[3]{r} - a_k\right| \leq c.$$

As output we need the approximated cube root and the smallest integer k for which the above happens.

▶ *specifications & notes* :

1. Define custom function for recursive computation.
2. Use `fabs(` *_double_* `)` from `math.h` for absolute value.
3. Use `cbrt(` *_double_* `)` from `math.h` for cube root.

8.8 Problem Set 08 - Recursive functions in C-II

1. **Sum of Digits**
   By now, you are well aware about the problem of computing the sum of digits of a positive integer. Write a program in C, which takes a positive integer as an input and computes the sum of its digits both recursively & iteratively.
   ▶ *specifications & notes* :
   1. Define two custom functions, one each for the recurence computation and closed form evaluation.

2. **Viete Method** ⋆
   In 1593, François Viète published a way to express the reciprocal of π as the following infinite product of nested radicals:

$$\frac{1}{\pi} = \frac{1}{2} \cdot \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdots = \frac{1}{2}\prod_{n=0}^{\infty} \frac{a_i}{2},$$

$$a_i = \begin{cases} \sqrt{2} & i = 0, \\ \sqrt{2 + a_{i-1}} & otherwise. \end{cases}$$

   Write a program in C programming language which takes a positive real c $(0 < c < 1)$ as input from the user and recursively approximates $\frac{1}{\pi}$ (which will give π_{approx}) untill we have

$$\left|\pi_{actual} - \pi_{approx}\right| \leq c.$$

   As output we need the approximated π and the smallest last index i for which the above happens.
   ▶ *specifications & notes* :
   1. Define custom function for recursive computation.
   2. Use `fabs( _double_ )` from `math.h` for absolute value.
   3. Use `sqrt( _double_ )` from `math.h` for square root.
   4. Use `M_PI` from `math.h` for actual π.
   5. Since the recursion formula is being used for each $i = 0, 1, 2 \ldots k$, is there a way to use the value from the pervious step in the current step?

3. **Reverse an array**
   By now, you are well aware about the problem of reversing an integer. Write a program in C, which takes a positive integer as an input and computes its reverse recursively only.
   ▶ *specifications & notes* :
   1. Define a custom function, for the recurence computation.
   2. **Do not** use `pow(_double_, _double_)` from `math.h` .

8.9 Problem Set 09 - Simulating probability distributions in C using rand( )

1. Write a program in C programming language to simulate the **'single-coin tossing experiment'** for an unbiased coin. Clearly, the sample space is $\Omega = \{H, T\}$ and if this experiment realises the probability distribution $\mathbb{P} : \Omega \to [0, 1]$ then your code must statistically verify that $\mathbb{P}$ is the uniform probability distribution.

2. Write a program in C programming language to simulate the **'single-coin tossing experiment'** for a *biased coin*. The *bias* is such that, H appears twice the number of times T appears. Clearly, the sample space is $\Omega = \{H, T\}$ and if this experiment realises the probability distribution $\mathbb{P} : \Omega \to [0, 1]$ then your code must statistically verify that $\mathbb{P}$ is **not** the uniform probability distribution.

3. Write a program in C programming language to simulate the **'single-dice throwing experiment'** for an unbiased dice. Clearly, the sample space is $\Omega = \{1, 2, 3, 4, 5, 6\}$.

   (a) If this experiment realises the probability distribution $\mathbb{P} : \Omega \to [0, 1]$ then your code must statistically verify that $\mathbb{P}$ is the uniform probability distribution.

   (b) If $A \subseteq \Omega$ is the event defined as $A \coloneqq \{\omega \in \Omega \mid \omega = 0 \ mod \ 2\}$, then compute $\mathbb{P}(A)$ using your code.

   (c) If $B \subseteq \Omega$ is the event defined as $B \coloneqq \{\omega \in \Omega \mid \omega \leq 4\}$, then compute $\mathbb{P}_A(B)$ and $\mathbb{P}_B(A)$.

   (d) If $C \subseteq \Omega$ is the event defined as $C \coloneqq \{\omega \in \Omega \mid \omega = 1 \ mod \ 2\}$, then compute $\mathbb{P}(A \cap C)$ using your code and thus verify that events A and C are mutually disjoint events.

4. Consider the **'double-die throwing experiment'** for two unbiased die, i.e., two unbiased die are thrown simultaneously. If $\Omega = \{1, 2, 3, 4, 5, 6\}$, then the sample space is $\Omega \times \Omega$. Write a program in C programming language to simulate this experiment and then use your code to compute the following, when this experiment realises the probability distribution $\mathbb{P} : \Omega \times \Omega \to [0, 1]$. $\star$

   (a) If $A \subseteq \Omega \times \Omega$ is an event such that $A \coloneqq \{(s, t) \in \Omega \times \Omega \mid s + t = 0 \ mod \ 2\}$, then compute $\mathbb{P}(A)$.

   (b) If $B \subseteq \Omega \times \Omega$ is an event such that $B \coloneqq \{(s, t) \in \Omega \times \Omega \mid s + t = 0 \ mod \ 4\}$, then compute $\mathbb{P}(B)$ and numerically verify that $\mathbb{P}(B) \leq \mathbb{P}(A)$. Any theory explaining as why this is happening?

5. Consider experiment Π named **'Chausar'**, where two 4-sided die are thrown simultaneously. Die-01 has marked 1, 2, 3, 4 whereas die-02 has faces marked 5, 6, 7, 8. The sample space is $\Omega_1 \times \Omega_2$ where $\Omega_1 = \{1, 2, 3, 4\}$ and $\Omega_2 = \{5, 6, 7, 8\}$. Write a program in C programming language to simulate Π and then use your code to compute the following, when Π realises the probability distribution $\mathbb{P} : \Omega_1 \times \Omega_2 \to [0, 1]$. In Π, we play Chausar using two unbiased die. $\star\star$

   (a) If $A \subseteq \Omega_1 \times \Omega_2$ is an event such that
   $A \coloneqq \{(s, t) \in \Omega_1 \times \Omega_2 \mid s \text{ and } t \text{ differs at most by } 3\}$, then compute $\mathbb{P}(A)$.

   (b) If $B \subseteq \Omega \times \Omega$ is an event such that $B \coloneqq \{(s, t) \in \Omega_1 \times \Omega_2 \mid s + t = 1 \ mod \ 2\}$, then compute $\mathbb{P}(B)$.

6. Consider Π_1, a **'random walk'** experiment, where a walk is performed along the X-axis, starting from 0. At each step:

   - the walker can either move forward by a unit distance with probability 3/4, or
   - the walker can move backward by a unit distance with probability 1/4.

   Write a program in C programming language which simulates this random walk Π_1, such that it takes a positive integer n as input and determines as on to which direction the walker ends after this random walk? What is the distance traversed? ⋆

7. Consider Π_2, a **'random walk'** experiment, where a walk is performed in the 1^{st} quadrant of the XY-plane, starting from 0. At each step:

   - the walker can either move along the Y-axis by a unit distance with probability 1/2, or
   - the walker can move along the X-axis by a unit distance with probability 1/2.

   Write a program in C programming language which simulates this random walk Π_2, such that it takes a positive integer n as input and determines on which point in the XY-plane the walker ends after this random walk? The output must be coordinates of a point in the XY-plane. Is it counter-intuitive that the x and y coordinate of the final point is always the same? ⋆

8.10 Problem Set 10 - Matrices & Graphs in C

Problems based on graph categorisation

1. Write a program in C programming language which takes the *adjacency matrix* of a general undirected graph G as input (element-wise). The program should detect the presence of any loops in G, and if any it must count the number of loops.

2. Write a program in C programming language which takes the *adjacency matrix* of a simple undirected graph G as input (element-wise). The program must count the total number of edges and the sum of the degrees of all the vertices in G. Incorporate relevant results you have seen in the Theory of Graphs.

3. Write a program in C programming language which takes the *adjacency matrix* of an undirected multigraph G as input (element-wise). The product must return the count of the total number of multi-edges present in G.

4. Write a program in C programming language which takes the *adjacency matrix* of a general graph G as input (element-wise). The program must detect whether the corresponding graph G is a directed graph or an undirected graph. ⋆

Problems based on DFS

1. Write a program in C programming language which takes the number vertices, the number of edges and the edges (vertex pairs) of a tree as inputs. For a given starting vertex, the program must perform a DFS to find the last vertex of the tree. The output must be the path traversed during the search. ⋆

2. Write a program in C which counts the number of connected components for a given graph, using a DFS. ⋆

3. Write a program in C which detects cycles in a given undirected graph, using a DFS approach. ⋆⋆

Ideas

Now that you have seen how to generate *pseudo-random* integers in C using the `rand()` function, consider the following program (8.10) which generates a random integer matrix, equivalent to a typical *adjacency matrix* for an undirected graph G, which does not have multi-edges. Can this concept be implemented in the above listed programs?

```c

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int main()
{
        srand(time(NULL));

        int n;
        printf("Enter the order of the graph: ");
        scanf("%d", &n);

        int G[n][n];

        for(int i=0; i < n; i++)
        {
                for(int j=0; j < n; j++)
                {
                        //...scope for meaningful experimentations
                        G[i][j] = rand() % 2;

                }
        }

        printf("\n");
        printf("The generated random matrix is :\n");

        for(int i=0; i < n; i++)
        {
                for(int j=0; j < n; j++)
                {
                        printf("%d ", G[i][j]);
                }
                printf("\n");
        }

        return 0;
}
```

EXTRAS

9.1 Problem Set 08 - Asymptotics-I (solutions)

1. Find the $\Theta-$bounds for the following recurrences.

   (a) $T(n) = 4T(n/2) + c$ where $T(1) = c_0$

$$\begin{aligned} T(n) &= 4T(n/2) + c \\ 4T(n/2) &= 16T(n/4) + 4c \\ 16T(n/4) &= 64T(n/8) + 16c \\ 64T(n/8) &= 256T(n/16) + 64c \\ &\vdots \\ 4^{k-2}T(n/2^{k-2}) &= 4^{k-1}T(n/2^{k-1}) + 4^{k-2}c \\ 4^{k-1}T(n/2^{k-1}) &= 4^{k}T(n/2^{k}) + 4^{k-1}c \end{aligned}$$

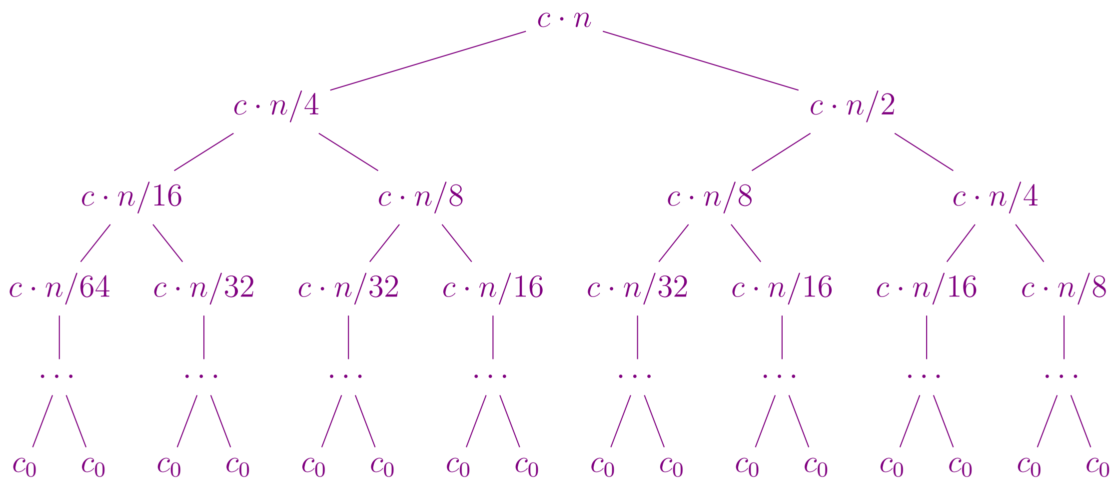
Let n such that, $n/2^k = 1$. Then $n = 2^k$ and,

$$\rightarrow \log_2 n = k \times \log_2 2 = k$$

$$\rightarrow 4^{k-1}T(n/2^{k-1}) = 4^{k}c_0 + 4^{k-1}c$$

Now, add all the recurrences, and cancel out all terms except T(n), $4^k c_0$ and the terms accounting for the constant cost c for each branching of the recurrence tree.
$T(n) = 4^{\log_2 n}c_0 + c(4 + 16 + 64 + \cdots + 4^{\log_2 -1})$
$\Rightarrow T(n) = n^{\log_2 4}c_0 + c(\frac{4(4^{\log_2 n-1}-1)}{4-1})$
$\Rightarrow T(n) = n^2 c_0 + c(\frac{n^{\log_2 4}-4}{3})$
$\Rightarrow T(n) = n^2 c_0 + \frac{c(n^2)}{3} - \frac{4c}{3}$
$\Rightarrow T(n) = n^2(\frac{3c_0+c}{3}) - \frac{4c}{3}$
$\Rightarrow T(n) \in \Theta(n^2)$

   (b) $T(n) = T(n/4) + T(n/2) + n \cdot c$ where $T(1) = c_0$



The cost of each level in the recursion tree is as follows:

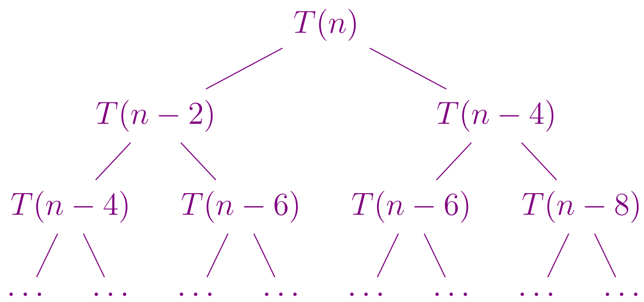
1. $c \cdot n$

2. $c \cdot \frac{n}{4} + c \cdot \frac{n}{2} = \frac{3cn}{4}$
3. $c \cdot \frac{n}{16} + c \cdot \frac{n}{8} + c \cdot \frac{n}{8} + c \cdot \frac{n}{4} = \frac{9cn}{16}$
4. $c \cdot \frac{n}{16} + c \cdot \frac{n}{32} + c \cdot \frac{n}{32} + c \cdot \frac{n}{16} + c \cdot \frac{n}{32} + c \cdot \frac{n}{16} + c \cdot \frac{n}{16} + c \cdot \frac{n}{8} = \frac{27cn}{64}$

We see that at each level the work is being multiplied by $\frac{3}{4}$.
(**Exercise** : Prove this with Induction).
Thus, the total work done is,

$$\begin{aligned}
\sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i c \cdot n &= cn + \frac{3cn}{4} + \frac{9cn}{16} + \dots \\
&= \frac{cn}{1 - \frac{3}{4}} && \text{(GP Sum)} \\
&= 4cn \\
&\in \Theta(n) && (c_0 = \tfrac{1}{4(c+1)},\ c_1 = 5(c+1))
\end{aligned}$$

(c) $T(n) = T(n-2) + T(n-4)$, where $T(0) = T(1) = T(2) = T(3) = c_0$.



We have : $T(n) = T(n-2) + T(n-4)$.
$n = 2k \ \Rightarrow T(2k) = T(2(k-1)) + T(2(k-2))$
Let $a_k = T(2k)$
The resulting recurrence is: $a_k = a_{k-1} + a_{k-2}$
Using the fibonacci recurrence : $\Rightarrow a_k \in \Theta(\varphi^n)$
Now, $T(0) = T(1) = T(2) = T(3) = c_0$
$\Rightarrow \frac{\varphi^n}{1+c_0} \leq T(n) \leq (1 + c_0)\varphi^n$
$\Rightarrow T(n) \in \Theta(\varphi^n)$

(d) $T(n) = T(n-1) + T(n-2) + k$, where $T(0) = 0$ and $T(1) = 1$.

2. Solve the following linear-homogeneous recurrences and comment on their $O-$bounds.

(a) $F(n) = 7F(n-1) - 12F(n-2),\ n \geq 2$ and $F(0) = 5, F(1) = -5$.
$F(n) = 25.3^n - 20.4^n$

(b) $F(n) = F(n-1) + 2F(n-2),\ n \geq 3$ and $F(1) = 0, F(2) = 6$.
$F(n) = 2.(-1)^n + 2^n$

(c) $F(n) = -F(n-1) + 4F(n-2) + 4F(n-3),\ n \geq 3$ and $F(0) = 8, F(1) = 6, F(2) = 26$.
$F(n) = 2.(-1)^n + (-2)^n + 5.2^n$

(d) $F(n) = 4F(n-1) - 4F(n-2),\ n \geq 3$ and $F(1) = 1, F(2) = 3$.
$F(n) = (1 + n).2^{n-2}$

(e) $F(n) = 8F(n-1) - 16F(n-4)$, $n \geq 4$ and $F(0) = 1, F(1) = 4, F(2) = 28, F(3) = 32$.
$F(n) = (1+2n).2^n + n.(-2)^n$

(f) $F(n) = -3F(n-1) - 3F(n-2) - F(n-3)$, $n \geq 3$ and $F(0) = 1, F(1) = -2, F(2) = -1$.
$F(n) = (1+3n-2n^2).(-1)^n$

3. For function f, g and h mapping from $\mathbb{N}$ to $\mathbb{R}^+$, prove the following:

   (a) If $f = O(h)$ and $g = O(h)$, then $f + g = O(h)$, where $f + g : \mathbb{N} \to \mathbb{R}^+$ defined as $\big(f+g\big)(x) = f(x) + g(x)$.

   (b) If $f = O(h)$ and $g = O(h)$, then $f \cdot g = O(h)$, where $f \cdot g : \mathbb{N} \to \mathbb{R}^+$ defined as $\big(f \cdot g\big)(x) = f(x)g(x)$.

4. For $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, show that $p(x) = O(x^n)$.

5. Prove the following asymptotic identities for $n \in \mathbb{N}$:

   (a) $n! = O(n^n)$.
   $1 \leq n;\ 2 \leq n;\ 3 \leq n;\ 4 \leq n; \cdots;\ n \leq n$. Thus we get :
   $1 \cdot 2 \cdot 3 \cdots \cdot n \leq \underbrace{n \cdot n \cdot n \cdots \cdot n}_{n\ times}$
   $\Rightarrow n! \leq n^n$.
   Thus for $c = 1$ and $n_0 = 1$ we have $n! = O(n^n)$.

   (b) $\log n! = O(n \log n)$.
   From part(a) we borrow $n! \leq n^n$
   $\Rightarrow \log n! \leq n \log n$.
   Thus for $c = 1$ and $n_0 = 1$ we have $\log n! = O(n \log n)$.

   (c) $\ln n = O(n)$.

6. Prove that for some $k \in \mathbb{N}$, $\sum_{i=1}^{k} i^8 = O(k^9)$. ⋆
   $1^8 \leq k;\ 2^8 \leq k;\ 3^8 \leq k;\ 4^8 \leq k; \cdots;\ k^8 \leq k$.
   $\Rightarrow 1^8 + 2^8 + \cdots + k^8 \leq \underbrace{k^8 + k^8 + \cdots + k^8}_{k\ times} = k^9$.
   Thus for $c = 1$ and $n_0 = 1$ we have $\sum_{i=1}^{k} i^8 = O(k^9)$.

7. Prove that for some $k \in \mathbb{N}$, $\sum_{i=1}^{k} i^8 = \Omega(k^9)$. ⋆⋆⋆
   Observe that $\sum_{i=1}^{k} i^8 = \underbrace{\sum_{i=1}^{\lceil k/2 \rceil - 1} i^8}_{I} + \underbrace{\sum_{i=\lceil k/2 \rceil}^{k} i^8}_{II}$.
   In summation II, each term is at least $\big(\lceil k/2 \rceil\big)^8$. Further the number of terms in summation II is $k - \lfloor k/2 \rfloor$ which is approximately $k/2$.
   Hence, $\sum_{i=1}^{k} i^8 \geq \frac{k}{2} \cdot \big(\lceil \frac{k}{2} \rceil\big)^8 \geq \frac{k}{2} \cdot \big(\frac{k}{2}\big)^8 = \frac{k^9}{2^9}$.
   Thus for $c = 2^{-9}$ and $n_0 = 1$ we have $\sum_{i=1}^{k} i^8 = o(k^9)$.

8. Prove that $n^2 + 17n = n^2 + o(n \ln n)$. ⋆⋆⋆
   To prove that $n^2 + 17n = n^2 + o(n \ln n)$ boils down to show that $n = o(n \ln n)$.
   Observe that $\lim_{n \to \infty} \frac{n}{n \ln n} = \lim_{n \to \infty} \frac{1}{\ln n} = 0$. By definition, for $c = 1$ and $n_0 = 1$ we have $n = o(n \ln n)$. This completes the proof. *qed*

9.2 Problem Set 12 - Graphs-I (solutions)

To test command over basic definitions & notations

1. Consider a *directed* graph G. Prove that graph G being strongly connected implies that G is weakly connected, and not the other way around.
Exercise.

2. Consider a *directed* graph G. Prove that graph G being a Null graph implies that G is also an Empty graph. Provide a counter-example to show that the other way around is not always true.
Exercise.

3. The general graph shown in the following figure(4) goes by the name of *GraphBuster* in standard literature. Count and determine the cardinality of V and E in *GraphBuster*.

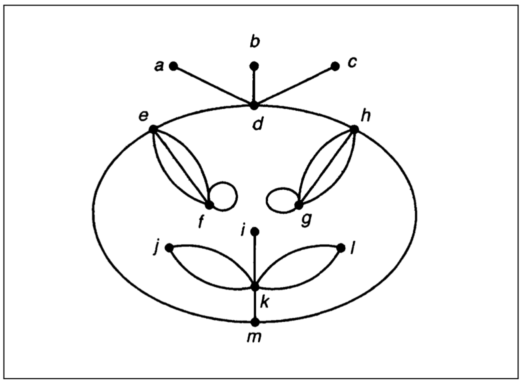


Figure 4: GraphBuster

GraphBuster has order order 13 with 21 edges.

4. A graph of order n is called *complete*, denoted by K_n provided that each pair of distinct vertices forms an edge. Show that a complete graph of order n has $n(n-1)/2$ edges.
Number edges equals total posible ways to choose 2 vertices from the n vertices in k_n, i.e., $\binom{n}{2} = \frac{n(n-1)}{2}$. □

Problems of type ¬ (simple)

5. Let G be a general graph. Show that the sum of the degrees of all the vertices of G is an even number, and consequently, the number of vertices of G with odd degree is even.
Each edge in G contributes 2 to the overall sum of the degrees in G. Hence, sum of all the degrees in G is $2e$, where is the total number of edges in G. Clearly, $2e$ is even. Further, let e_e and e_o be the number of vertices with even and odd degrees respectively. e_o is difference of two even numbers, hence it is even. □

6. If G is a simple graph of order $n \geq 3$, such that for all pairs of distinct vertices x and y in G that are not adjacent, we have $deg(x) + deg(y) \geq n$, then show that G must be connected. ⋆
Proof by contradiction. Suppose that G is not connected. We then show that G cannot satisfy the Ore property : $\forall$ pairs of distinct vertices x and y that are not adjacent, $deg(x)+deg(y) \geq n$. Since G is not connected, its vertices can be partitioned

into two parts, U and W, in such a way that there are no edges joining a vertex in U with a vertex in W. Let r be the number of vertices in U and let s be the number of verticed in W. Then $r + s = n$, and each vertex in U has degree at most $r - 1$, and each vertex in W has degree at most $s - 1$. Let x be any vertex in U and let y be any vertex in W. Then x and y are not adjacent, but the sum of their degrees is, at most, $(r - 1) + (s - 1) = r + s - 2 = n - 2$, and this contradicts the Ore property. We conclude that if G satisfies the Ore property, then G must be connected. $\square$

7. Let G be the graph such that elements of $\{1, 2, 3, \cdots, 20\}$ form its vertices. In G, two vertices (integers) are joined by an edge if and only if their difference is an odd integer. Show that G is a bipartite graph.
   Let $X = \{k \in \mathbb{N} \mid k = 1 \ mod \ 2; k \leq 20\}$ and $Y = \{k \in \mathbb{N} \mid k = 0 \ mod \ 2; k \leq 20\}$. Observe that X and Y together is a bipartition for the graph G. $\square$

8. Prove that if a multigraph G is bipartite, then each of its cycles has even length. *Note that: length of any cycle/path is the number of edges it is composed of.*
   G is a bipartite multigraph with bipartition X, Y. The vertices of a walk of G must alternate between X and Y. Since a cycle is closed, this implies that a cycle contains as many left vertices as it does right vertices and hence has even length. $\square$

9. For a fixed $n \in \mathbb{N}$, let $\mathrm{G_n}$ be the graph such that elements of $\{0, 1\}^n$, the set of all $n-$length binary strings, form its vertices. In $\mathrm{G_n}$ any two vertices are joined by an edge if and only if they differ in exactly one 1-bit. Show that G is a bipartite graph. $\star$
   Proof. Let n be a positive integer. We consider the set of all $n-$tuples of 0s and 1s as the vertices of a graph Q_n with two vertices joined by an edge if and only if they differ in exactly one coordinate. If $X = (x_1, \cdots, x_n)$ and $Y = (y_1, \cdots, y_n)$ are joined by an edge, then the number of 1s in Y is either one more or one less than the number of 1s in X. Let X consist of those $n-$tuples that have an even number of 1s; let Y consist of those $n-$tuples that have an odd number of 1s. Then two distinct vertices in X differ in at least two coordinates and hence are not adjacent. Similarly, two distinct vertices in Y are not adjacent. Hence, Q_n is a bipartite graph with bipartition X, Y. $\square$

10. Prove that a graph of order n with at least

$$\frac{(n-2)(n-1)}{2} + 1$$

   edges must be connected.
   Proof by contradiction. Assume that G is a disconnected graph of order n such that the number of edges is at least $\frac{(n-2)(n-1)}{2} + 1$. WLOG, say there are only two components in G such that one component is a singleton vertex and the other is of order $n - 1$. It follows that, the non-trivial component can have $\frac{(n-1)(n-2)}{2}$ edges at max whenever it is complete. Observe that, in that case the number of edges that G would have is $\frac{(n-1)(n-2)}{2}$, which is a contradiction to the primal assumption of our proof. Therefore, G is connected. $\square$

11. Prove that a graph of order n with every vertex having degree at least $\frac{n}{2}$ must be connected.
   Proof by contradiction. We borrow the construction/set-up of a graph G exactly from the previous proof. Observe that, once again the non-trivial component would have a degree $(n - 2)/2$ at max, which is lesser than $n/2$. Hence, a contradiction. $\square$

12. In a simple graph if two vertices x and y are joined by a path then, show that they are also joined by a simple path.
Exercise.

9.3 Problem Set 13 - Graphs-II (solutions)

Problems of type (simple)

1. The two graphs shown below in figure (5) have the same number of vertices and edges. Prove that despite these they are not isomorphic.

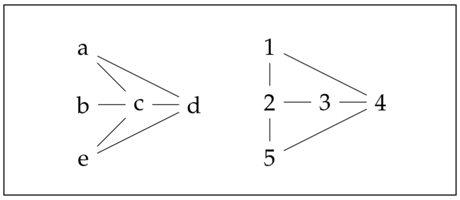


Figure 5: Sample graphs 01

$deg(b) = 1$ in the first graph, but is no vertex in the second graph with this property.

2. Consider the two graphs shown below in figure (6), where both the graphs have same degree sequence $(3, 3, 3, 3, 3, 3)$. Show that despite this, they are not isomorphic.

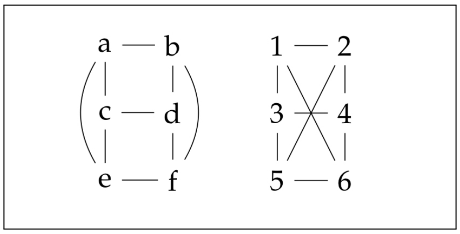


Figure 6: Sample graphs 02

Note that (a, c, e) forms a 'triangle' in that any two pairs are adjacent. There is no such triangle in the second graph, so these graphs are not isomorphic.

3. A graph has 26 vertices and 58 edges. There are five vertices of degree 4, six vertices of degree 5, and seven vertices of degree 6. If the remaining vertices all have the same degree, what is this degree?
There are $26 - 5 - 6 - 7 = 8$ vertices of degree x. Applying Euler's Theorem, we get $5 \cdot 4 + 6 \cdot 5 + 7 \cdot 6 + 8 \cdot x = 2 \cdot 58$. On solving, $x = 3$.

4. A graph has 24 vertices and 30 edges. It has five vertices of degree 4, seven pendant vertices, and seven vertices of degree 2. All other vertices have degree 3 or 4. How many vertices of degree 4 are there?
Let x be the number of vertices of degree 3, and y the number of vertices of degree 4. The order of the graph is 24 therefore : $5 + 7 + 7 + x + y = 24$. After applying Euler's Theorem, we get $5 \cdot 4 + 7 \cdot 1 + 7 \cdot 2 + 3 \cdot x + 4 \cdot y = 3 \cdot 30$. On solving, $y = 4$.

Problems of type $\neg$ (simple)

5. Prove or Disprove, whether a bipartite graph can have K_3 as its subgraph?
A bipartite graph cannot have a K_3 subgraph, as there would be three mutually adjacent vertices in the graph, requiring three different partite sets, contradicting that the graph is bipartite.

6. Let $d_0(G)$ be the least among the degrees of the vertices of an $n-$vertex graph G. Prove that if $d_0(G) \geq (n-1)/2$, then the graph G is connected. ⋆
   Proof by contradiction : Assume that graph G is not connected and $d_0(G) \geq (n-1)/2$. Let one component have m vertices, where $1 \leq m \leq \lfloor n/2 \rfloor$. Then the maximum degree of any vertex in that component is at most $m-1$, since it is only connected to vertices within that component. Therefore, at least one vertex in G has degree $\lfloor n/2 \rfloor - 1 < (n-1)/2$. This contradicts the fact that $d_0(G) \geq (n-1)/2$, hence G is connected.

7. Prove that a graph G with v vertices and e edges has at least $v-e$ connected components. [Hint : use induction on e.]
   BASE CASE : $e = 0$; then each of the v vertices of G are connected components in their own right. Hence, number of connected components $= v - 0$.
   INDUCTIVE HYPOTHESIS : $e = k$, the graph G of order v has $v - k$ connected components.
   INDUCTIVE STEP : $e = k+1$; consider the Graph G from the IH, where all k edges have been encompases by its connected components. Hence, for the additional edge, we will have to joing two connected components. It follows, that due to the additional edge there will be a decrease in the number of connected components, so the resultant graph has $v - k - 1 = v - (k+1)$ connected components.

8. Prove that a connected graph G with n vertices contains at least $n-1$ edges. [Hint : the proof might be an application of the result in the previous question, when proved!]
   Any connected graph G, trivially has 1 connected component. Hence the proof is a direct application of the result in question 7.

9. If G is a connected graph with v vertices and e edges, then $v \leq e+1$.
   Proof by contradiction : Let us assume that graph G is a connected graph of order v with e edges, such that $v > e+1$. Since G is connected, there must be at least one edge corresponding to each vertex. Considering this, $v > e+1$ constraint is a contradiction to G being connected. Hence $v \leq e+1$.

10. If G is a connected graph, then removing an edge from a cycle will not make G a disconnect graph. ⋆
    Arbitrarily select a vertices v_i, v_j, part of a cycle in G. Any cycle starting from and ending on a vertex v_i, which includes v_j is of the form :

$$v_i - v_{k_1}, v_{k_1} - v_{k_2}, v_{k_2} - v_{k_3}, \cdots, v_{k_m} - v_j, v_j - v_{l_n}, \cdots, v_{l_1} - v_i,$$

    such that $[v_i - v_{k_1}, \cdots, v_{k_m} - v_j]$ and $[v_j - v_{l_n}, \cdots, v_{l_1} - v_i]$ are two distinct simple paths between the vertices v_0 and v_n. Thus removing an edge from the cycle is equivalent to remove an edge from either of the two simple paths. The other one remains intact connecting v_0 and v_n. This completes the proof. □

9.4 Problem Set 14 - Graphs-III (solutions)

1. A graph $G = (V, E)$ is called k-regular if $\mathsf{deg}(v) = k$ for all $v \in V$. A graph is called regular if it is k-regular for some k. Give example of a regular bipartite graph.
   All complete bipartite graphs are regular, whenever the order of the bipartitions are equal. **Food for thought:** can be there other examples?

2. Prove that every induced subgraph of a complete graph is complete.
   A *complete graph* K_n is a graph where every pair of distinct vertices is connected by an edge. An *induced subgraph* H of a graph G is formed by taking a subset of vertices $V(H) \subseteq V(G)$, and including all edges from G that exist between these vertices.

   Let $G = K_n$ be a complete graph with vertex set V. Let H be an induced subgraph of G, formed by a subset $V(H) \subseteq V$. By the definition of an induced subgraph:

   > For every pair of vertices $u, v \in V(H)$, the edge $uv \in E(H)$ if and only if $uv \in E(G)$.

   Since G is complete, every pair $u, v \in V(G)$ is connected by an edge in G. Therefore, for any $u, v \in V(H)$, the edge $uv \in E(G)$, and thus it must also be in $E(H)$. Hence, every pair of distinct vertices in H is connected, and so H is a complete graph. $\square$

3. Prove that every subgraph of a bipartite graph is bipartite.
   Proof by contradiction. Let G be a bipartite graph with bipartitions X, Y. We consider a subgraph $G_1 \subseteq G$, such that G_1 is not complete. We borrow the result that subgraph of a bipartite graph is again bipartite. It follows that, $\exists\ v_x \in X$ and $v_y \in Y$ which are not adjacent. But this is a contradiction as G is assumed to be complete. This is forms *good-enough* sketch of the complete proof. $\square$

4. If two graphs G_1 and G_2 are isomorphic then their degree sequences are the same.
   Suppose $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic graphs. Let $f : V_1 \to V_2$ be the isomorphism between them. Since f preserves adjacency, the number of neighbors of any vertex $v \in V_1$ is equal to the number of neighbors of $f(v) \in V_2$. That is,

   $$\deg_{G_1}(v) = \deg_{G_2}(f(v)).$$

   Thus, for every vertex in G_1, there exists a corresponding vertex in G_2 with the same degree. Therefore, the set (or multiset) of degrees in G_1 is identical to that in G_2. Therefore, the degree sequences of G_1 and G_2 are the same. $\square$

5. What is the sum of the entries in a row of the adjacency matrix of an undirected simple graph?
   Degree of the vertex corresponding to that row.

6. Let u, v, and w be three distinct vertices in a graph. There is a path between u and v and also there is a path between v and w. Prove that there is a path between u and w.
   Let p_1 be the path connecting u and v; p_2 be the path connecting v and w. Then the resultant path $p_1 \cup p_2$ connects u and w. $\square$

7. Suppose $(d_1, \ldots, d_n)$ be a degree sequence of a tree. Determine $\sum_{i=1}^{n} d_i$.
   Exercise.

8. Show that the number of vertices n in a binary tree is always odd.
   A binary tree is a tree in which every node has at most two children. For a *full binary tree* (i.e., every internal node has exactly two children), there is a well-known relationship, $n = 2i + 1$ where i is the number of internal nodes.

   Each internal node has exactly 2 children. So, the total number of children is $2i$. Each node, except the root, is a child of some node. Therefore, the total number of nodes n is : $n = \text{internal nodes} + \text{leaves} = i + (i + 1) = 2i + 1$. Since $2i + 1$ is always odd for any integer i, we conclude that the number of vertices n in a full binary tree is always odd. □

9. Let p be the number of pendant vertices in a binary tree T with n vertices. Show that

$$p = \frac{n+1}{2}.$$

   Exercise.

10. Let $k \in \mathbb{N}$ be the height of a binary tree T. Determine the maximum number of leaf nodes of T.
    Claim: The number of nodes at level k of a full binary tree is 2^k, where root node is present at level 0. Prove the claim using induction. Hence, the maximum number of leaf nodes of T, whenever $k \in \mathbb{N}$ is the height of a binary tree T equals to 2^k. □

11. Consider the graph defined by the adjacency matrix provided below.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ & & 0 & 0 & 1 & 0 & 1 & 0 \\ & & & 0 & 0 & 1 & 0 & 1 \\ & & & & 0 & 0 & 1 & 1 \\ & & & & & 0 & 1 & 1 \\ & & & & & & 0 & 1 \\ & & & & & & & 0 \end{bmatrix}$$

    (a) Determine if it is an Euler graph.
    Exercise.

    (b) Determine if it admits a Hamiltonian circuit.
    Exercise.

    (c) Give a spanning tree of this graph.
    Exercise.

Notes : Course Review

1. This section is under construction and the appropriate content shall be made available after the formal termination of the course.
2. Stay tuned!