Logic:- is Algebra:-

↓

very fundamental.

Expert Systems:-

(1980's concept)

(old school AI)..

repository of "facts" to answer querys

Basic notions:-

1) **Proposition:-** a statement, either true of false.

↓ -refer to things in "Domain of disclousre"

may simply refer to

property of a domain.

Predicate

Eg: Alice has wings.

an eg: characters of 'alice in wonderland'!

Domain = {alice, jabberwock}.

Predicate: "the part of sentence, containing a verb, telling something about the object".

2) **Predicate:**

a function, that assigns True or False to each element of "Domain of disclosure".

Eg:

	winged } - predicate
Alice	F
Jabberwock	T
flamingo	T

domain of disclosuer.

predicate (element) is a proposition. (True or False).

* Propositional calculus ≡ Boolean Algebra.

1) we've got operators (Eg: ~~"operate on"~~).

unary operator: (i) Negation ($\neg p$)

ii) Or ($p \vee q$)

iii) and ($p \wedge q$)

iv) if p then q ($p \rightarrow q$)

Same as ($\neg p \vee q$)

& - and, conjunction.

'or' - or, disjunction.

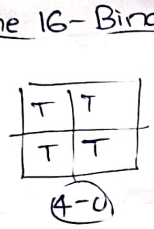
$\neg$ ~ negation.

p \ q	T	F
T	T	F
F	T	T

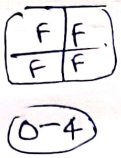
$2 \times 2 \times 2 \times 2$

16 Binary operators to be strict!

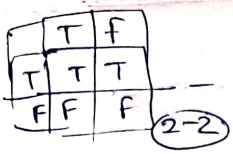
the 16- Binary operators:



(T) (4-0)

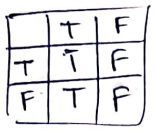


(F) (0-4)

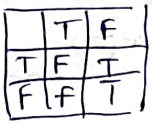


(1) (2-2)

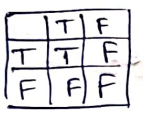
→ ¬(1)



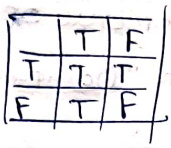
(2)



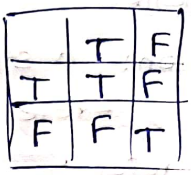
→ ¬(2)



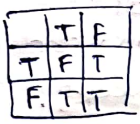
∧ (3-1) (1-3) and



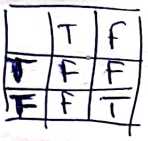
∨ OR



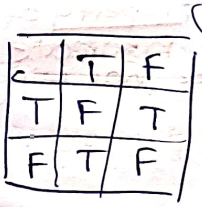
↔ iff



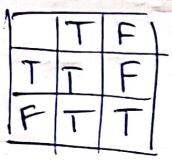
↑ NAND



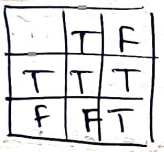
↓ NOR



(2-2) ⊕ XOR.



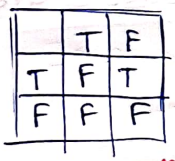
→ implies



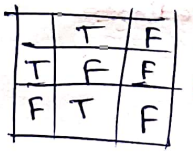
← implied by.

iff ≡ (p→q) ∧ (q→p)

~~bi-relation~~ equivalent statements p & q are.



↛ not implies (101)



↚ not implied by

* p → q

p implies q. ✓

if p, then q. ✓

~~p if q~~

p only if q. | p iff q = (p if q) ∧ (p only if q)

(p ← q) ∧ (p → q)

---

if not q ~~then~~ not p

not p if not q.

} contrapositive form.

¬q → ¬p

Note: p

p if q

(p

* Some e

1) cont

2) Distri

a) De mor

¬(

¬(

4) No

Note: p iff q

p if q $(p \leftarrow q)$

p only if q $(p \rightarrow q)$ (not english intuitive).

You lazily, thought reversed

p if q and ~~not p if not q~~ $\equiv$ if not q then not p.

$$p \rightarrow q \equiv \neg q \rightarrow \neg p$$

* Some equivalencies:-

1) contrapositive:-

$p \rightarrow q \equiv (\neg q) \rightarrow (\neg p)$ the english statements make absolute sense.

2) Distributive:- (even for more than 2, in brackets)

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

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

$p \wedge p = p$

$p \vee p = p$

3) Demorgan laws:- (apply repetedly for more than 2 terms).

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

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

} operators are exchanged.

4) No order significance:-

$(p_1 \wedge p_2 \wedge p_3 \wedge p_4 \dots) \rightarrow$ can be evaluated in any order.

$(p_1 \vee p_2 \vee p_3 \vee p_4 \dots) \rightarrow$ can be evaluated in any order.

Think in english.... (not always!

we are trying to make an "ALZEBRA" Here).

* propositional calculus also has

quantifiers: → to what quantity predicate evaluates to TRUE.

$\forall x \ldots, \exists x \ldots$

Eg: My proposition:

1) All characters in Alice in w... are Winged

Truth value = (F)

Notation: $\forall x$ Winged(x). → a proposition!

Here, we dont see truth of "Winged(x)".

we see truth of "$\forall x$ Winged(x)".

"Is true" is the default to check, for P(x) (or) $\forall x P(x)$ says that P(x) is true for all x.

the definition.

Now! examine its truth!

2) Some character in Alw is winged.

$\exists x$ Winged(x) → this complete is a proposition! (a complex one!).

NOTE:-

universal quantifiers

$\forall x f(x) = f(x_1) \wedge f(x_2) \wedge f(x_3) \wedge \ldots$

$\exists x f(x) = f(x_1) \vee f(x_2) \vee f(x_3) \vee \ldots$

existential quantifiers..

members!

Alzebra part.

(can't always think in english!).

* $\neg(\forall x f(x)) = \exists x (\neg f(x))$

$\neg(\exists x f(x)) = \forall x (\neg f(x))$

english proof:- Intutive!

Alzebra proof:-

$\neg(f(x_1) \wedge f(x_2) \wedge \ldots)$

$= \neg f(x_1) \vee \neg f(x_2) \vee \ldots$

$= \exists x \neg f(x)$

* Moving the quantifiers; in case of Predicate with 2 args.

- $\forall x \forall y J(x,y) \equiv \forall y \forall x J(x,y)$

  a predicate; should be true.

  a predicate; should be true

  a predicate; you need to check this.

  But not $\forall x \exists y P(x,y) \equiv \exists y \forall x P(x,y)$.

- $\exists x \exists y P(x,y) \equiv \exists y \exists x p(x,y)$

- R is a proposition, independent of x.

  $\forall x P(x) \vee R = (\forall x P(x)) \vee R$.

  Scope is like

  $\forall x, (P(x) \vee R)$

  this whole is a predicate over domain.

  only P(x) is a predicate over Domain

  $\neg(\exists x P(x))$ is same $\forall x \neg P(x)$.

  * Don't think in English, you will go MAD!

  * Think in ALGEBRA.

$$\forall x (P(x) \vee R) = (P(x_1) \vee R) \wedge (P(x_2) \vee R) \wedge \cdots$$

(associativity law)

$$= (P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge \cdots) \vee R$$

$$= (\forall x P(x)) \vee R$$

Hence.

$$\forall x (P(x) \wedge R) = (\forall x P(x)) \wedge R$$
$$\forall x (P(x) \vee R) = (\forall x P(x)) \vee R$$
$$\exists x (P(x) \wedge R) = (\exists x P(x)) \wedge R$$
$$\exists x (P(x) \vee R) = (\exists x P(x)) \vee R.$$

} don't at all think english.

Big Fool:-

$$(\forall x P(x)) \vee (\forall x Q(x)) \equiv (\forall x P(x)) \vee (\forall y Q(y))$$

the x, works inside the bracket only!

these both 'x' are not related.

variable change, doesn't change shit.

$$\equiv (\forall x P(x)) \vee R.$$

doesn't depend on x...

So;

$(\forall x\, P(x)) \vee (\forall x\, Q(x)) \equiv \forall x \forall y\, (P(x) \vee Q(y))$

Interchangeable ✓

↳ ain't same Both are "their own x!"

↳ mind you!

---

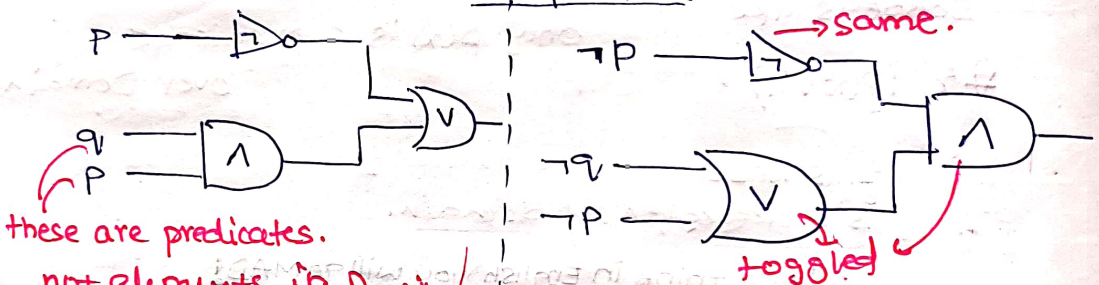
the looking mirror:- (Huge Demorgan).

$\sim f(p,q) = f'(\neg p, \neg q)$

↳ in f, ∨ & ∧ are toggled

¬ is kept intact.

for operators:-



these are predicates. not elements in Domain.

negating in mirror.

for quantifiers:-

predicate, member

$\forall x\, P(x)$ → $\exists x\, \neg P(x)$

$\exists x\, P(x)$ → $\forall x\, \neg P(x)$

Two quantifiers

$\forall x \exists y\, Likes(x,y)$ → $\exists x \forall y\, \neg Likes(x,y)$

** Like de-morgans:-

$\neg(P \_\_ Q) \equiv (\neg P \_\_ \neg Q)$

- ★ And ($\wedge$) —— or ($\vee$)
- XOR ($\oplus$) —— iff ($\leftrightarrow$)
- NAND ($\uparrow$) —— NOR ($\downarrow$)
- implies ($\rightarrow$) —— not-implied by ($\not\leftarrow$)
- implied by ($\leftarrow$) —— not implies ($\not\rightarrow$)

take $P = \neg a$
$\neg P = a$ equals
$(\neg a) \vee \neg a$
$\neg(\neg a) \vee \neg a$
$a \vee \neg a$
$\neg P \leftarrow \neg Q$
$\neg P \vee \neg P$
$\neg(\neg P \vee Q)$
$(a \vee \neg P)$
$(P \vee \neg Q)$

Proof:

$\neg(P \rightarrow Q) \equiv (\neg P \_\_\_ \neg Q)$

$\neg(P \Rightarrow Q)$
$\neg(\neg P \vee Q)$
$P \wedge \neg Q$
$\neg(\neg(\neg P) \wedge \neg Q)$

$\neg$

	T	F
T	T	F
F	T	T

	T	F
T	F	T
F	F	F

$\rightarrow$

P \ $\neg Q$	F	T
F	F	T
T	F	F

$\equiv$

$\neg P$ \ $\neg Q$	T	F
T	F	F
F	T	F

$\equiv \neg$

	T	F
T	T	T
F	F	T

$\neg(q \rightarrow p)$.

$\neg(P \leftarrow Q)$
not implied by ✓.

(or):

using $\neg f(P,Q) = f'(\neg P, \neg Q)$
↳ toggle $\wedge, \vee$
intact $\neg$.

$\neg(P \rightarrow Q)$

$\neg(\neg P \vee Q) = \neg(\neg P) \wedge \neg(Q)$
$= P \wedge (\neg Q)$
$= \neg(\neg P \vee Q)$
$= \neg(\neg P \vee \neg(\neg Q))$
$= \neg(\neg P \leftarrow \neg Q)$ ✓

** Vacuous truth:-

a conditional statement is only true, because the antecedent cannot be satisfied. (coz, precedent is false).

Eg: "all cellphones are switched off" will be true even if there are no cellphones

$a \rightarrow b$ is true; if a is false.

we say $a \rightarrow b$ is vacuously true.

Proofs: → ONLY for the Propositions which are true; but can't use this fact, in proof.
→ Trying to mechanize.

- we are proving propositions.
  often called Lemma, theorems, claims.
- stated using various predicates, specific to the system.
  already specified as definition.

Eg: all positive even natural numbers are greater than 1.

$$\forall x \ (\text{positive}(x) \wedge \text{even}(x)) \rightarrow \text{greater}(x,1)$$

* Here there are also axioms for a system: (truth value is TRUE).
  define the system. $\forall x \ x+0=x$

A proof's anatomy:-

1) clearly state the proposition to be proved.
2) derive propositions $P_1, P_2, \ldots P_k$ where; for each k,
   a) P_k is axiom for the system.
      (or)
   b) P_k is a proven proposition. (~~it has only 1 valid truth~~ value) in the system.
      (or)
   c) if $(P_1 \wedge P_2 \wedge P_3 \ldots \wedge P_{k-1}) \rightarrow P_k$ is true; then also you are allowed to use P_k.

- Note that;
  if $(P_1 \wedge P_2) \rightarrow P_k$ is true;
  then $(P_1 \wedge P_2 \wedge P_3 \ldots) \rightarrow P_k$ is true!.
  since ( ) is enough! to imply P_k..
  (or, verify by thinking of truth values).

- Example:

definition: An integer x is said to be odd; iff there exists an integer y s.t. $x = 2y+1$

↓ a proposition; with truth value = TRUE

$$\forall_{\in Z} x \ \text{odd}(x) \leftrightarrow \exists_{\in Z} y \ (x = 2y+1)$$ → a proposition; with truth value: true

P_1

Proposition: (proof statement)

$$\forall x \in Z \ \ \text{odd}(x) \rightarrow \text{odd}(x^2)$$

need to prove. ✓ (PTO)

→ proof:-

let x be an arbitrary ~~statement~~ element in Z.

humancreativity; to decide which P_i to start with.

* suppose x is odd. (if x is even; the proposition $odd(x) \rightarrow odd(x^2)$ is vacuously true).

i.e. as odd(x) is F;
() → () is T.

by P_1:

$\exists a \quad x = 2a+1$

$\therefore x^2 = 4a^2+4a+1$

$x^2 = 2(2a^2+2a)+1$

} arithematic propositions which are proved already.

$\exists b \in Z \ (x^2 = 2b+1) \leftrightarrow odd(x^2)$ by P_1.

$\therefore$ if odd(x) = TRUE, then odd(x^2) = TRUE

hence

$odd(x) \rightarrow odd(x^2)$ is TRUE; "Proved".

(this long proof; to mechanize the process of proof).

all cleverness goes into writing the proof.

not verifying it.

---

* multiple approaches:-

1) direct deduction. (1 step procedure)

- rewriting the proposition. (in contrapositive)
- proof by contradiction (we take false & encounter contradiction)
- proof by giving example (counter-example)
- mathematical induction.

proof templates:-

1) $P \to Q$:

then our arsenal of true propositions

Start with P_0.

i.e.

$P_0 = P$

P_1

P_2

:

bcoz; its suppose 'P'

i.e.

if P = false;

$P \to Q$ is vacuously true.

Hence;

now consider

P = true.

& maybe $P \Rightarrow Q$

(or)

$P \Rightarrow P_1 \Rightarrow P_2 \cdots \Rightarrow Q$

Note:

- "Barber cuts hair of those"

is a proposition.

Gimme a predicate.

① cuthair (B, x) → two inputs

② cuthair by B (x). → 1 input.

- "who dont cut their own hair"

is another proposition.

Gime a predicate.

① $\neg$ cuthair (x, x)

② $\neg$ cuthair themselves (x)

→ here if we choose ①,①; life easy.

→ if ②,② again we need new predicates P_1, P_2, P_3 (similar to ①)

2) Rewriting the proposition:-

- contrapositive form:-

prove $(P \to Q) \equiv \neg Q \to \neg P$.

so;

$(P_0 \wedge P_1 \wedge P_2 \cdots) \Rightarrow \neg Q \to \neg P$

$\Downarrow$

$P \to Q$.

Eg:

$\forall x, y \in Z^+ \left[\underbrace{x \cdot y > 25}_{P} \to \underbrace{(x \geq 6) \vee (y \geq 6)}_{Q} \right]$

heh; we prove by take $x < 6 \,\&\, y < 6$

$\therefore x \cdot y \leq 25$. ($P \to Q$ template again)

i.e.

$\neg Q \to \neg P$

$\therefore P \to Q$

- proof by contradiction:-

lets say;

we need to prove proposition 'P'.

not the $P \to Q$ form now.

$P \equiv \neg P \to$ false

$\equiv \neg(\neg P) \vee F$

$\equiv (P \vee F)$

$\equiv P$ waw!

Why this makes life easy? 'coz, now we got a $P_0 (= \neg P)$

say P is false, & then; prove!

"False" → this also a proposition which ~~may be true~~ or false?

in english:-

take P to not hold & show a contradiction. i.e. "False"

hence P is true. (Here, eat my shit!)

* $\neg P \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \cdots P_n \Rightarrow$ False

$n > 3$

proof templates:-

3) reduction:

prove p.

so,

show $(r \to p)$ & prove r.

reduces the task; to prove r.

(thanks to the work done in showing $(r \to p)$)

so; one mathematician proves $(r \to p)$ & other, proves r. HOPE

essentially; science is this way.

One man didn't prove wave nature of light.

1 man didn't prove photoelectric effect.

6) cases: (case analysis)

to prove q; say cases are — c_1, c_2, c_3

prove:

- $(c_1 \vee c_2 \vee c_3)$ atleast one of three cases holds.
- $c_1 \to q$
- $c_2 \to q$
- $c_3 \to q$

thus we have "proven"

$(c_1 \vee c_2 \vee c_3) \to q \equiv (c_1 \to q) \wedge (c_2 \to q) \wedge (c_3 \to q)$

& also $(c_1 \vee c_2 \vee c_3)$ is true

$\therefore q$ is true.

think english:-

$c_1 \to q$, $c_2 \to q$, $c_3 \to q$

& always, atleast one of c_1, c_2, c_3 holds. $\therefore q$ is true.

4) template for $\exists x\, p(x)$:- & $\neg(\forall x\, p(x))$ proof by counter example. → show $(\exists x \neg p(x))$

proof by example. ✓

elaborately;

if I need to prove

$T \equiv \exists x\, p(x) \to q(x)$

1) example for $p(x) =$ False. (asshole!)

proposition T is true (vacuously true)

2) example for $q(x) =$ True DONE!

proposition T is true (irrespective of $p(x)$)

3) example for

$(\neg p(x) \vee q(x))$ is true

instead of specifically choosing one of these.

Might not find one x;

but show that it exists.

(like; not able to find satwik; but find his shit & conclude that satwik exists) !

5) Template for $\forall x\, p(x)$:

"let x be an arbitrary element in domain of p.

work hard for $p(x)$.

$\therefore$ since x is arbitrary;

$\forall x\, p(x)$ ✓

Q) Which of the following are valid approaches for showing $\exists x \neg Q(x)$?

a) Show $\forall x Q(x)$

b) Show for some x, $\neg Q(x)$ holds

c) Show that $\neg \exists x Q(x)$.

d) Show $\forall x \neg Q(x)$. Now, now... don't be an ass...

this "shows" _____?

THINK ONCE.

★ cases eg: $\forall a,b,c,d \in Z^+$ if $a^2+b^2+c^2=d^2$, then { d is even iff a,b,c are even }

Sol) $P_0 \equiv a^2+b^2+c^2=d^2$. (true). (∵ if false, propotion is vacuously true).

a,b,c are even $\iff$ d is even.

r q.

$r \to q$ is direct.

let c_1: a,b,c are all odd.

c_2: a,b,c are, 1 even, 2 odd.

c_3: a,b,c are 2 even, 1 odd.

& $(c_1 \vee c_2 \vee c_3) \equiv \neg r$

if $(c_1 \vee c_2 \vee c_3)$ is true, q is false

$q \to (c_1 \vee c_2 \vee c_3 \vee r)$ (T)

$(q \to c_1) \vee (q \to c_2) \vee (q \to c_3) \vee (q \to x)$

show these three are false.

"proof by contradiction".

$\therefore$ $q \to r$ is true.

Proof by Induction:-

proof by programming. (lol)

* prove: $\forall x \in \mathbb{Z}^+ \; P(x)$.

soln: - 1st, prove P(1) (Base case)

- next prove the proposition P: $\forall k \in \mathbb{Z}^+ \; P(k) \rightarrow P(k+1)$

weak-Mathematical Induction.

∴ we conclude; P(1) → P(2)

P(2) → P(3)

⋮

induced... & P(n) is true ∀n.

* so:

to prove $\forall x \in \mathbb{Z}^* \; P(n)$

- we show P(1) & $\forall k \in \mathbb{Z}^+$ P(k) → P(k+1)
- Then by (weak) mathematical induction; $\forall x \in \mathbb{Z}^+ P(x)$

In disguise:-

- To prove $\forall n \in \mathbb{Z}^+ P(n)$:
  - prove P(1) & show $\forall k \in \mathbb{Z}^+ \neg P(k+1) \rightarrow \neg P(k)$
  - so; finally proved.

via weak induction;
when we wanna prove P(k+1), even though we know P(1), P(2), ... P(n) are all true; we cannot use this as unless, u add them to basecases. this "was supposed to be proved by weak Ind. itself!

→ Strong induction:-

- to prove $\forall n \in \mathbb{Z}^+ P(n)$:
  - prove P(1)
  - Show $\forall n \in \mathbb{Z}^+ (P(1) \wedge P(2) \wedge \cdots P(n)) \rightarrow P(n+1)$

(same as weak induction.) - Then why!

In this proof; we have a large number of supposed to hold true propositions; to prove P(n+1). Strong!

* Importance of Strong mathematical induction:-

Both are equivalent; just; By using weak; we need to increase the

Showing $\forall k \in z^+ \; (P(1) \wedge P(2) \wedge \cdots P(k)) \rightarrow P(k+1)$ rather than Showing $\forall k \in z^+ \; P(k) \rightarrow P(k+1)$.

Q) prove: every positive integer $n \geq 2$, there exists some prime factorization.

(its unique! not proved here)

Strong induction:-

Base case: let $n=2$.

then $n = 2^1$.

True.

NOW:

Show $(P(2) \wedge P(3) \wedge \cdots P(k)) \rightarrow P(k+1)$.

So; $n = k+1$

if n is prime:- proved ✓ as RHS is TRUE

if n is composite:-

By definition

$\exists a \in [2, k] \; a \mid k+1$

$\therefore$ $k+1 = a \times (b)$ $b < k+1$

now; suppose $(P(2) \wedge P(3) \cdots P(k))$ is (T)

then $P(2), P(3), \cdots P(k)$ are (T)

this, in weak ind.; must be included in basecases.

i.e. 2, 3, 4, ... a, ... K can be factorized.

$\therefore$ K+1 is factorizable.

Proved induction statement.

Hence proved.

so; again we fall to strong ind. indirectly.....

weak induction:-

Base case: $n=2$.

$\therefore$ $P(2)$ holds.

NOW:

Show $P(k) \rightarrow P(k+1)$.

if K+1 is prime: okay

if K+1 is composite:-

$k+1 = a \times (b)$

where $a, b < k+1$.

Now what! You "still" dont know whether P(a), P(b) hold.

Suppose P(k) is true. — & dance!

So; we now need to show before hand, a, b are factorizable

i.e. they should be in basecases also.

* How do we include all $n < K$ in basecases?

technical issue, removed in strong by following strong ind. logic.

(don't use $P(k) \rightarrow P(k+1)$ in proof of itself)

Numbers:-

$\mathbb{Z} = \{-2,-1,0,1,\dots\}$

$\mathbb{N}$ = natural number $\{0,1,2,3,\dots\}$

↳ so that idiotic quiz question is fine.

$\mathbb{Z}^+ = \{1,2,3,\dots\}$

(1) $\forall n \in \mathbb{Z}$ $n|0$

true by defn $n = q \cdot d$

& $0|0$

Since $0 = (1)\cdot 0$

q exists!

definition of divisibility

$d|n \Rightarrow$

$\exists q \in \mathbb{Z}$ s.t. $n = q \cdot d$

d is divisor ✓

factor ✓

$d(12) = \{\pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 12\}$

* GCD (0,10) = 10.

so; $gcd(a,b) \leq \min(|a|,|b|)$ is false.

* $gcd(-12,-18) = 6$

don't write -6.

$|0|$.

→ * $gcd(a,b) = gcd(a, b+na)$

$= gcd(a, gcd(a,b))$

gives ~~gauss~~ GCD euclid method.

$q \triangleq gcd$

meaning "defined as"

* The one dimensional lattice:-

$L(a,b) = \{ua+bv \mid u,v \in \mathbb{Z}\}$

then $L(a,b)$ is {all multiples of $gcd(a,b)$}

i.e.

a lattice;

with edge length = $gcd(a,b)$

(regular arrangement.)

Names:

euclids lemma:-

if a prime number divides product of 2 nos.,
then it divides one of the two numbers.

Euclids algorithm:-

for gcd:
do repeated division.

EEA:-

Bezout's identity:

$\exists \{u,v\} \in \mathbb{Z}$ such that

$a \cdot u + b \cdot v = \gcd(a,b)$ for $a,b \in \mathbb{Z}$

repeated division runs, you can get one such $\{u,v\}$;

this is EEA.!

(extension, to just finding gcd by repeated division)

well-ordering:-

there exists a minimum, in a finite subset of integers.

fundamental theorem of arithmetic:-

every no. is prime factorizable, in a unique manner.

Q-R theorem:

$a = q \cdot b + r$; $r < |b|$ & $r \geq 0$.

→ modular arthematic:-

$a \equiv b \pmod m$ implies $m \mid a-b$.

generally take $m > 1$

*

$a \equiv b \pmod 0$ means $a = b$

$a \equiv b \pmod 1$ means a,b can be anything.

→ $a \equiv b \pmod m \iff \text{rem}(a,m) = \text{rem}(b,m)$

remainder $\in [0, m)$.

(r)

$=-2$	-7	-6	-5	-4	-3	-2	-1	
$=-1$	0	1	2	3	4	5	6	→ $\pmod 7$
$q=0$	7	8	9	10	11	12	13	
$=1$	14	15	16	17	18	19	20	

lie in same column

$10 \equiv 17 \pmod 7$

$4 \equiv -3 \pmod 7$.

$[a]_m = \{x \mid a \equiv x \pmod m\}$

∴ the set of $\mathbb{Z}$ is broken into m columns; by congruency operator.

1) fix an 'm'.

universe is made finite

2) let $[a]_m$ stand for the column containing a.

i.e. all x s.t. $a \equiv x \pmod m$

$[-17]_5 = [-2]_5 = [3]_5$ } many representations for the same "object."

∴ $\mathbb{Z}_m = \{[0]_m, [1]_m, [2]_m, \cdots [m-1]_m\}$

operations:- on columns. not on individual numbers.

1) modular addition:-

$[a]_m +_m [b]_m \triangleq [a+b]_m$

"defined as"

- inherits all properties
  - identity
  - inverse
  - commutative
  - associative.

for m = 5

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

2) modular multiplication:

$$[a]_m \times_m [b]_m \triangleq [a \cdot b]_m$$

"defined as."

* $[1]_m$ → identity of multiplication.

commutative ✓
associative ✓

multiplicative inverse? → product of two non-zeros can be zero.

Eg.: $[m]_5$

$\times_m$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

} all have m. inverse.

$\times_m$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

NO m-inverse.

* a has a multiplicative inverse modulo m;

if $\gcd(a,m) = 1$.

proof:

$\gcd(a,m) = 1$

$\exists\, u, v$ s.t.

$au + mv = 1$

$\therefore [a]_m \times [u]_m = [1]_m$

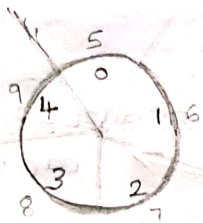
* if $[a]_m \times [u]_m = [1]_m$; then $\gcd(a,m) = 1$, $\gcd(u,m) = 1$

$[0]_m$; $m > 1$ → doesn't have multiplicative inverse. (101).

$\gcd(0,m) = m$! terrific.

→ modular arithematic; on a clock:

essentially (mod m) na!



m = 5.

skippyclock.

say; skip = 'n'.

Q1) when does hour-hand can possibly reach all digits?

Ans: when gcd(n,m) = 1 (sufficient & necessary)

i.e. ∃ t, v s.t. nt + mv = 1

nt: t times the pointer skipped

mv: to get a reminder (valid digit on clock)

1: reminder

• if gcd(n,m) = 2 (say);

then nt − mv = 1

never happens.

Sufficency:-

Say

nt − v·m = 1

then

n(2t) − m(2v) = 2

okay! ✓

Q2) ---what if gcd ≠ 1?

* also if gcd(n,m) = g (≠1)

Say n = gp

m = gq

make smaller clock;

q digits

&

pointer's skip is p.

-now; all digits reached!

∴ g(0), gx1, gx2, ··· gxq−1 are reached on main clock.

Q3) which hour is reached twices; for the first time?

sol) Hour-0 (or your start point!)

Bcoz; scenario is,

Hour-0 → Hour→q → Hour→b → c → 0 → a → b → c → 0 ···

Here, b ≠ 0; can't give rise to a; as function is one-one-onto. (if not m,n; for p,q)

Before giving any other answer; 0 comes first; repeates first.

→ Like;

→ Chinese reminder theorem:-

use to get common term of AP.

Skip = 1.

a digits → r = some element

b digits → s = some element

" if (a,b)=1, then for all posibilities (r,s); there exist a unique x; such that:-

$x \equiv r \pmod a$ & $x \equiv s \pmod b$ "

$r \in [0, a-1]$, $s \in [0, b-1]$

> with 0.9; $x \in [0, a \cdot b - 1]$; such x exists!!

↓ infinite x's

$x_1 = x + ab$

$x_2 = x + 2ab$

(not great)

(chinese reminder thm.)

(0,1) → $1 = bv$ → $1 - au$

$au + bv = 1$

$au \equiv 1 - bv$

$\therefore (m,n) = m(1-au) + n(1-bv)$

* CRT Representation:-

if (a,b)=1; represent $x \in Z_{ab}$ as

$(r,s) = (\text{rem}(x,a), \text{rem}(x,b)) \in Z_a \times Z_b$.

uniquely map from Z_{ab} to $Z_a \cdot Z_b$.

* if m=ab; arthematic in Z_m is done from arthematic in Z_a, Z_b.

addition → straight.

multiplication → straight.

Like

$x \equiv r \pmod a$

$x' \equiv r' \pmod a$

$xx' \equiv rr' \pmod a$

$x + x' \equiv r + r' \pmod a$.

$x_{ab} \equiv (r,s)$

•also find additive & multiplicative inverse.

$(r,s) +_{(ab)} (r',s') = (0,0)$

$(r,s) \times_{(ab)} (r',s') = (1,1)$

∴ r' is multi-inverse of r wrt a. ∴ if x's multiplicative inverse has to exist,

s' " " of s wrt b. (r, s)'s mult... inverses have to exist!

(to derive euler's totient function)

→ Like; 0 1 2 3 4

3 0 3 1 4 2

5 spaces are to be filled with 5 values (if gcd=1) so; no repetition scope.

CRT representation, for more than 2 factors:-

Suppose $m = a_1 \cdot a_2 \cdots a_n$;

where $gcd(a_i, a_j) = 1 \quad \forall\, i \neq j$

then for any $(r_1, r_2, \cdots r_n)$, $r_i \in [0, a_i)$; there exist unique solution x

in $[0, m)$ - for the systems of congruencies

$$x \equiv r_i \pmod{a_i} \quad \forall i = 1, 2, \cdots n$$

bcoz, onto & sizes are equal both sides; ∴ one-one.

one side	otherside
x in $[0, m)$	$(r_1, r_2, \cdots r_n)$ in $([0, a_1), [0, a_2), \cdots [0, a_n))$

apply concept like

a_1 ; $(a_2, a_3 \cdots a_n)$

then

a_2, $(a_3 \cdots a_n)$

then

a_3, $(a_4, a_5 \cdots a_n)$

---

* Z_m^* → elements of Z_m which have multiplicative inverse.

such elem. is called unit of Z_m

$Z_4^* = \{1, 3\}$

i.e. unit is co-prime with m.

* $len(Z_m^*)$ = no. of naturals nos. coprime with m; & less than m.

euler's totient funct^n.

* $len(Z_m^*) = m \cdot (1 - \frac{1}{p})$ if $m = p^k$ for $k = 1$ or 2 or ....

(solve by calculating; nos. with p as factor $\{p^{k-1}$ such nos.

$\therefore$ ans. $= p^k - p^{k-1}$

$= m(1 - \frac{1}{p})$

$x \equiv 5 \pmod 6$ — 5, 11, (17), 23, 29, ~~35~~

$x \equiv 3 \pmod 7$ — 3, 10, (17)

* How many units in Z_m?
  (defined before.)

> say $m = p_1^{d_1} \cdot p_2^{d_2} \cdots p_n^{d_n}$.

> now; x in Z_m is uniquely mapped to $(r_1, r_2, \cdots r_n)$ where $x \equiv r_i \pmod{p_i^{d_i}}$

> for x to be invertible, each of r_i in $Z_{p_i^{d_i}}$ is to be invertible.

> no. of units in $Z_{p_i^{d_i}} = p_i^{d_i}\left(1 - \frac{1}{p_i}\right)$ } multiplication rule.

∴ no. of units in Z_m = $p_1^{d_1}\left(1-\frac{1}{p_1}\right) \times p_2^{d_2}\left(1-\frac{1}{p_2}\right) \times \cdots$
('∵' one-one correspondance)

no. of units in $Z_m = m\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots$ my 7th class...

no. of units in Z_m ← $\boxed{\phi(m) = m\left(1-\frac{1}{p_1}\right)\cdot\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_n}\right)}$

- euler's ϕ function.
- Euler's totient function.

- if $(a,b)=1$; $\phi(ab) = \phi(a)\cdot\phi(b)$ } such fuctions are called multiplicative functions.

→ Little more structure:-

* if $a \in Z_m \setminus Z_m^*$ ($Z_m - Z_m^*$); then, $\exists u \in Z_m$ such that $au = 0$ & $u \neq 0$

take $u = \frac{m}{\gcd(a,m)}$ → don't take, this for multiplicative inverse.. lol...

$u < m$ & $u \neq 0$ ∴ > won't hold;

* if $a \in Z_m^*$ ∵ gcd() = 1 then.

if $a \in Z_m^*$ & $[au]_m = 0 \Rightarrow [u]_m = 0$ (taken!)

(or;) $\forall u \in Z_m, u \neq 0$; $au \neq 0$

2) $a \in Z_m^* \iff \bar{a}^{-1} \in Z_m^*$

because; $\bar{a}^{-1}$ has an inverse! ($=a$, ofc)

the cond^n to be in Z_m^*

3) $a, b \in Z_m^*$ then $ab \in Z_m^*$ (closed under multiplication)

$\because$ $b^{-1}.a^{-1}$ is inverse of ab. } the def^n; have a inverse, to be in Z_m^*.

automatically; gcd = 1.

4) for $a \in Z_m^*$

*: $a \cdot Z_m^* \triangleq \{ab \mid b \in Z_m^*\} = Z_m^*$

proof:

$ab \in Z_m^*$.

$\therefore a \cdot Z_m^* \subseteq Z_m^*$.

now; if $a \in Z_m^*$, $\bar{a} \in Z_m^*$.

$\therefore$ for some $x \in Z_m^*$

$\bar{a}^{-1} x \in Z_m^*$

multiply a on both sides.

$a \cdot \bar{a}^{-1} x \in a \cdot Z_m^*$

$x \in a \cdot Z_m^*$

$\Rightarrow Z_m^* \subseteq a \cdot Z_m^*$

$\Rightarrow Z_m^* \triangleq a \cdot Z_m^*$

→ modular exponentiation:-

- for $a \in Z_m$; $d \in Z^*$; $a^d \triangleq \underbrace{a \times_m a \times_m a \cdots}_{d \text{ times 'a'}}$

not modulo m.

- for $a \in Z$; $([a]_m)^d \triangleq [a^d]_m$

till now; d is positive.

expand; for $a \in Z_m^*$

- for $a \in Z_m^*$; $d \in Z$:

$a^0 = 1$; $a^d = a \times_m a \cdots$ if $d > 0$

$\bar{a}^d = (a^{-1})^d = (a^d)^{-1}$

$\because a^d \times (\bar{a})^d = 1$; $(a^d)^{-1} = (\bar{a})^d$

haha.

- $a^d \cdot a^e = a^{d+e}$, $(a^e)^d = a^{ed}$

Note: seem to subtly confuse multiplication & exponentiation.

$a \equiv 1 \pmod m$

$a^2 \equiv 1 \pmod m$ → sometimes, you write : $a^2 \equiv 2 \pmod m$ no!

clock, skipping: its $n \equiv ? \pmod m$; $n \cdot 2 \equiv ? \pmod m$ ... not $n^2 \equiv ? \pmod m$.

$a \cdot 2 \equiv 2 \pmod m$ ✓

- **euler's totient theorem:-**

for any $a \in Z_m^*$; $a^{\phi(m)} \equiv 1 \pmod m$

Logic: (WRONG!)

$\phi(m)$ elements in Z_m^*

$1 \; a_1 \; a_2 \; a_3 \cdots a_{\phi(m)}$

$\because a^0 = 1$

a ; say $= a_1$

a^2 would be some other elem...

a^3 would be SOME other elem...

:

all elements are covered

$\therefore a^{\phi(m)}$ is again 1.

→ (CORRECT)!

let u = product of all elem. in Z_m^*

w = " " " " in $a \cdot Z_m^*$

$\therefore u = \{\}$

$w = a^{\phi(m)} \cdot \{\}$

$\therefore w = a^{\phi(m)} \times u$:

but; $w = u$; as $a \cdot Z_m^* \triangleq Z_m^*$;

$\therefore a^{\phi(m)} = 1$ (multiply with u^{-1}; $u \in Z_m^*$ too!)

(in Z_m space ra!)

direct consequence

→ wrong!

$\{a^0, a^1, a^2, \cdots a^{\phi(m)}\}$

may not have all elem. of Z_m^*.

(if all are there; then a is generator of Z_m^*).

So; $\phi(m)$ might not be the smallest exponent such that $a^{\phi(m)} \equiv 1 \pmod m$

* put m = prime p;

$\forall a \in Z_p^*$; $a^{p-1} \equiv 1 \pmod p$ here, $(a,p)=1$

(or) say; fermat's little theorem.

$(a,p)=1$ $a^p \equiv a \pmod p$ for any a.] fermat's theorem.

→ cyclic structure of Z_p^* :-

now; take a clock; of size p;

which has multiplicative skip of g;

i.e. clocks starts at $g^0 \equiv 1$

&

$1 \to g \to g^2 \to g^3 \cdots \to g^{p-1}$

$\equiv 1$ } fermat's thm...

∴ in p-1 skips; hand comes to initial position again.

1 to g to g^2 to ....

$g \neq 0$

$g < p$

* Now; does there exist a 'g' such that all nos. are covered? (except 0)

Important fact: (couple pages long proof)

if p is prime; there exist 'some' g in Z_p^*; such that

$\{g, g^2, g^3, \cdots g^{p-1}\} \triangleq Z_p^*$ (equal to 1.)

obv $g \neq 1$; unless p=2. haha...

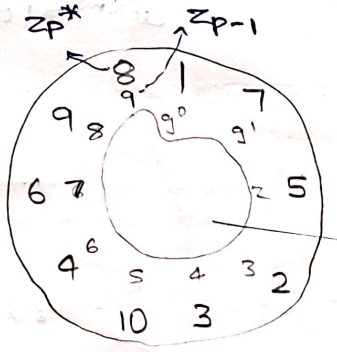
g is generator of Z_p^*.

need not be unique.

p=5; g=2,3

Eg:

take prime 11; g=7.



∴ $10 = g^{(5)}$ ~ discrete log.

$x = g^{(k)}$; $x \in Z_p^*$; $k \in Z_{p-1}$

> some how; we connected Z_p^* to Z_{p-1}.

for x in Z_{p-1}; $g^x \in Z_p^*$

↓ one-one.

& so, multiplication in Z_p^* is add^n in Z_{p-1}

easy to go from K to x; tough to go from x to K (brute force!)

cryptography: one way func: NP

→ more insigit into modular exponentiation:-

1) let $a \in Z_m^*$.

then $a^{\phi(m)} \equiv 1 \pmod m$

• use this, to compute a^x; for higher x.

let $x \equiv b \pmod{\phi(m)}$

then $a^x \equiv a^b \pmod m$

bcz, only $a \in Z_m^*$: $a^x = a^b$ in m-modwar.

2) now, let $y = x^e$; $\equiv [x]_m^{[e]_{\phi(m)}}$

& asked you to find e^{th} root of y.

Soln) → find b, such that $b \cdot e \equiv 1 \bmod (\phi(m))$ ($\because (e, \phi(m)) = 1$)

& $(x^e)^b \equiv x^{eb} \equiv x^1 \pmod m$]

∴ we found our x. ✓

Note: $a^{1/e} = b \leftrightarrow$ if $a = b^e$ | Hence; if $a \in Z_m^*$; $(a^{1/e})^d = (a)^d$ $\because [e]^{-1}_{\phi(m)} = [d]_{\phi(m)}$

might not exist; might not be unique.

There EXISTS unique b; if $(e, \phi(m)) = 1$

In these both cases; $(e, \phi(m)) \neq 1$ } don't blindly say, no e^{th} root.

& don't confuse multiplicat^n with exponentiat^n

multiplication; $a \cdot k = a \cdot (k \cdot m)$

exp^n: $a^k = a^{k - \phi(m)}$ when $a \in Z_m^*$ or $a \in Z_p$ (p prime) (coz; Z_p^* same to Z_p)

Limitation of our method.

3) m is product of primes. (two or more distinct). hence if $m = p^2 q$; not this case.

$m = pq$; where p & q are primes.

then

for a in Z_m no need of Z_m^*

$a^{\phi(m)+1} \equiv a \pmod m$ → fermat theorem.

write a; in CRT form;

• $(x, y) \} Z_m^*$

• $(x, 0) \}$

• $(0, y) \} Z - Z_m^*$

$a^{\phi(m)} \neq 1$ $a^{\phi(m)} \equiv 1$

But both, $a^{k\phi(m)+1} \equiv a \pmod m$

again; if $(e, \phi(m)) = 1$; $e \cdot d \equiv 1 \pmod{\phi(m)}$

$(a^e)^d \equiv a \pmod m \leftrightarrow a^{1/e} = a^d \pmod m$

to find e^{th} root.

now; $a \in Z_m$ where m = product of distinct primes.

$15^{1/2}$ in Z_{33} ? $\quad 33 = 3 \times 11$

sol) find; $15 = (0,4)$ in (Z_3, Z_{11})

$15^{1/2} = (0, 4^{1/2})$ $\quad 0^{1/2} = 0$

$\because$ unique soln; as 3 is prime.

$4^{1/2}$ in 11;

$\phi(11) = 10$; $(2,10) \neq 1$

$\therefore$ do manually;

$4^{1/2} = \pm 2$ in 11.

$\therefore 15^{1/2} = 9$ or 24 in Z_{33}.

* in Z_p^*; $(1)^{1/e}$ has exactly $\gcd(e, p-1)$ values.
* in Z_p^* $(a^e)^{1/e}$ has exactly $\gcd(e, p-1)$ values.

* for every $m > 2$; $\phi(m)$ is even.

$\therefore$ $\gcd(2, \phi(m)) = 2$.

unique sq.root doesn't exist for $m > 2$ elements.

- there is merit in discussing $a^{1/2}$ seperately.

→ Squaring & square root :-

Note $(a)^2 = (-a)^2$

& $a, -a$ are distinct for $m > 2$. (really?)

* element in Z_m has a square.

but many elements have more than one sq.rt.

$\therefore$ Some elements have no square roots.

* Quadratic residues :-

Elements in Z_m^* of the form x^2.

↓

not a prime?

hmmm.

• squares in Z_p^*: (p: prime)

$$Z_p^* = \{1, g, g^2, \cdots g^{p-1}\}$$

even-powers are only quadratic residues.

$$\therefore \left(\frac{p-1}{2}\right) = \text{length}(QR_p^*)$$

$$a^b = [a]_p^{[b]_{\phi(p)}}$$

* given z in Z_p^*, check if z is a square:

check $z^{\left(\frac{p-1}{2}\right)}$

if $\boxed{z^{\frac{p-1}{2}} \equiv 1}$; then z is square.

proof:

$z = g^{2k} \Rightarrow z^{\frac{p-1}{2}} = z^{k(\phi(p))} = 1$

$z = g^{2k+1} \Rightarrow z^{\frac{p-1}{2}} = g^{\frac{p-1}{2}} \neq 1$ $\because \frac{p-1}{2} \neq 0$ & g is generator;

* $(g^{2h})^{1/2} = g^h$ & $g^{h+\left(\frac{p-1}{2}\right)}$

if $\frac{p-1}{2}$ is even; parity of belong to QR is same So, gotta be different.

if $\frac{p-1}{2}$ is odd; parity is opposite.

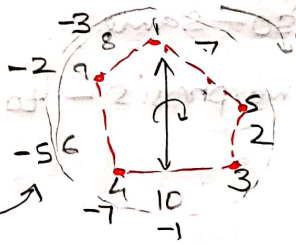
$\therefore (g^2)^{1/2} = g^1$ in $p=11$ is $(g^{2+10})^{1/2} = g^6$

* where are sqrts of elements in QR_p^* (say x^2)

$(1)^{1/2} = \pm 1$

$\boxed{g^{\frac{p-1}{2}} = -1}$ why? bcoz $g^{p-1} = 1$. & square root do!

(11)



generator.

• - QR

$g=7$

$\frac{p-1}{2} = 5 = \text{odd}$: exactly one of $\pm x$ in QR_p^*

(Hence; squaring is a permutation in QR_p^*)

$\frac{p-1}{2} = 6 = \text{even}$: both $\pm x$ in QR_p^*

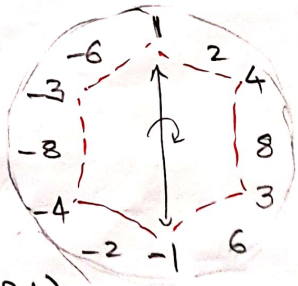
$\boxed{\sqrt{z} = z^{\frac{p+1}{4}}}$

$\frac{(p+1)}{4}$ → num is multiple of 4

$z^{\frac{p+1}{4}}$ & if $z = g^{2k}$;

then $z^{p+1/4} = g^{2(\text{some int})}$ // QR_p^*.

(13)



$g=2$

• - QR

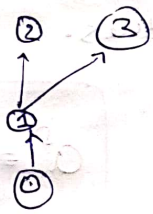
* diagonally opposite means:

$g^k, g^k \cdot g^{\frac{p-1}{2}}$

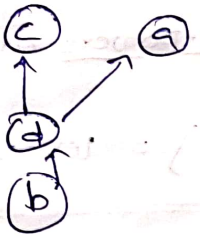
means $x, -x$ $(\because -1 = g^{\frac{p-1}{2}})$

For next-chapter: sets & relations:-

Relation R on {1,2,3,4} is



Relation R' on {a,b,c,d}



see! both are isomorphism.

the isomorphism function is, then wrt R, R'

0 — b
1 — d
2 — c
3 — a

* "isomorphism" is a bijection; which, when Function from S to S' $f(R) \equiv R'$ vaguely... like, apply on each (node) term of R.

↳ "isomorphic trees"

Etymology:
iso — same
morphosis — to shape / to form.

Sets:- !! set: unordered collection of objects:

* sets as predicates:

set, & its predicate.

x	winged(x):
Alice	false
Jabberwock	True
Flamingo	True

is winged = {Jabberwock, Flamingo}

• set operations:-

if S is set; $x \in S$ is a predicate.

Set op.	Propositional calculus:
$A \cup B$:	$x \in A \vee x \in B$
$A \cap B$:	$x \in A \wedge x \in B$.
$A - B$:	$x \in A \wedge \neg(x \in B)$
$A \Delta B$: symmetric difference	$(x \in A) \oplus (x \in B)$ XOR

$= (A-B) \cup (B-A)$

demorgans law:-

$\overline{S \cup T} = \overline{S} \cap \overline{T}$

$\neg(x \in A \vee x \in B) = \neg(x \in A) \wedge \neg(x \in B)$

set inclusion:-

* $A \subseteq B$

$x \in A \longrightarrow x \in B$

* $A \subseteq B \longrightarrow \overline{B} \subseteq \overline{A}$

$(\because (x \in A \rightarrow x \in B)$

$\equiv$

$\neg(x \in B) \rightarrow \neg(x \in A)$)

contrapositive

* Proving set equity:-

to prove $L = M$;

we do $L \subseteq M$

$M \subseteq L$.

Eg: $L = \{x \mid \exists u, v \text{ s.t. } x = au + bv\}$

$M = \{x \mid \gcd(a,b) \mid x\}$

show $L = M$.

$\vee\vee$

Sol) show $L \subseteq M$ & $M \subseteq L$.

* Inclusion-Exclusion:-

$n(A \cup B \cup C)$

$= n(A) + n(B) + n(C)$

$- n(A \cap B) - n(B \cap C)$

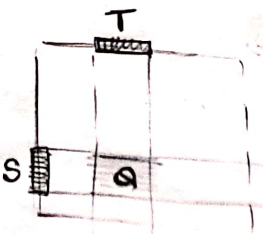
$- n(C \cap A)$

$+ n(A \cap B \cap C)$

* cartesian product:-

$Q = S \times T$ means

$Q = \{(s,t) \mid s \in S, t \in T\}$



$(\overline{S \times T}) = (\overline{S} \times \overline{T}) \cup (S \times \overline{T}) \cup (\overline{S} \times T)$

* if $a \times b = c \times d$;

then $a = c$ & $b = d$ only when, none of those is ϕ.

In general

$S \times T = T \times S \not\longrightarrow S = T$

(what if $S = \phi$)

* Relation:-

> A subset of SxS is called a relation.

> A predicate over the domain SxS is called relation.

Eg: is greater(x,y) → denoted as $x \boxed{>} y$ (a symbol to symbolize rel^n)

is relative(x,y)

x [rel.] y → some Symbol.

- 3 ways (out of many) to look at relations:-

(i) Ordered pairs:-

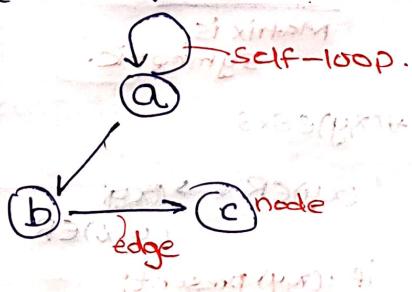
$R \subseteq S \times S$;

$R = \{(a,b), (a,a), (b,c)\}$

(ii) Boolean matrix:-

	a	b	c
a	T	T	F
b	F	F	T
c	F	F	F

(iii) Graph:-



* operations on Relations:-

> since relations is a set:

$R \cup R_1$

$R \cap R_1$

$\overline{R}$

⋮

> some operations; based on (a,b) structure of relations:-

1) Converse (a.k.a. transpose):-

$R^T = \{(y,x) : (x,y) \in R\}$

Bool. matrix is transposed too!

2) Composition:-

→ reverse convention for functions.

$R \circ R_1 = \{(x,y) \mid \exists w \in S \ (x,w) \in R \text{ and } (w,y) \in R_1\}$

if atleast one 'w' exists; then write (x,y)

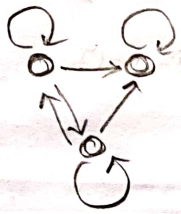
Bool matrices are multiplied;

addition ≡ 'or'

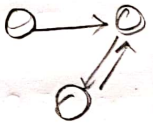
multiplicat^n ≡ 'and'

→ Categories of Relations:- (a relation; can be in two or more categories da. subset.)

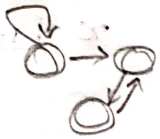
1) Reflexive
all should be there!
$\forall x \in S;\ (x,x) \in R$



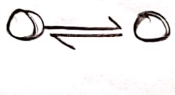
Irreflexive:-
$\forall x \in S,\ (x,x) \notin R$
(no self loops)



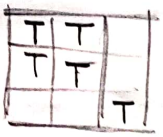
Non-reflexive.
Neither:-
for some $x \in S$,
$(x,x) \notin R$



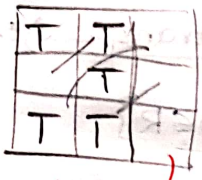
2) Symmetric:-
Metrix is symmetric.
$\forall (x,y) \in S \times S$
$(x,y) \in R \leftrightarrow (y,x) \in R$
if (x,y) present; (y,x) present.



(or)

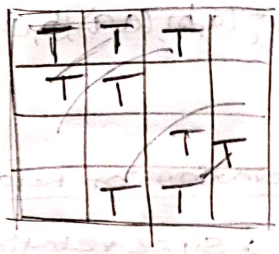


Anti-Symmetric:-
$\forall (x,y) \in S \times S$ &
$x \neq y$
$(x,y) \in R \rightarrow (y,x) \notin R$
- not even for one such (x,y) there should be (y,x) (leaving out $x=y$ cases)

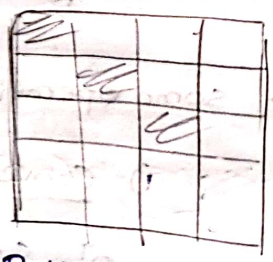


symmetric; need not necessarily mean reflexive.

Non-symmetric.
Neither:-



Both:



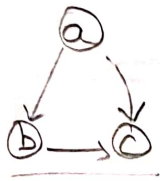
Both symmetric & antisymm.
for $x \neq y$
$(x,y) \in R$
$(y,x) \notin R$
Vacuously true.

$(x,y) \in R$
$(y,x) \in R$

asymmetric relation:-
an irreflexive - antisymmetric relation is termed asymmetric.

3) Transitive relation:-

> if (a,b) & (b,c) then we should have (a,c)



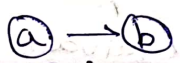
transitive.

∴ $R \circ R \subseteq R$

or, even

$R^k \subseteq R \quad \forall k > 1$

* if you find path from a to b (over multiple edges), then



exists.

- Intransitive:-

not transitive.

Antitransitive → not discussed ie.

* The relat^n set SxS is reflexive, symmetric, transitive; ∵ it has all possible edges & self loops.

*

Reflexive closure:- (of R) the minimal relation (subset of SxS), $R' \supseteq R$; such that R' is reflexive.

even if one elem removed, not reflexive

Symm. closure:- (of R) the minimal relation $R' \supseteq R$; such that R' is symm.

Transitive closure:- (of R) the minimal relation $R' \supseteq R$; such that R' is transitive.

* For a relation R; there exist a unique refl^x. closure, a unique symm. closure, a unique trans. closure.

(Steps to obtain the above closures;

> just keep on adding the required edges or self-loops).

bcoz; we take the minimal.

4) Equivalence Relation:

A relation; which is
- reflexive
- symmetric
- transitive.

Eg: "both have last dig = __"

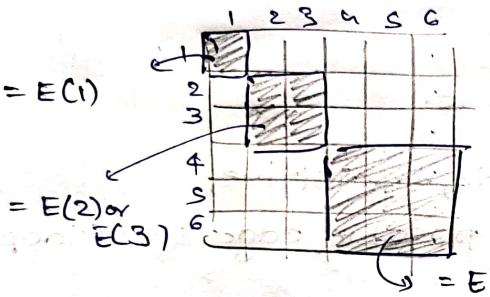
"congruency to mod 7"

* A equivalence relation (a subset of SxS) is partitioned into equivalence classes... → $P_1, P_2, \dots P_T$

$P_i \cap P_j = \emptyset$

$P_1 \cup P_2 \cup \dots P_T = E$

: $Eq(x) \triangleq \{y : x \sim y\}$



= E(1)

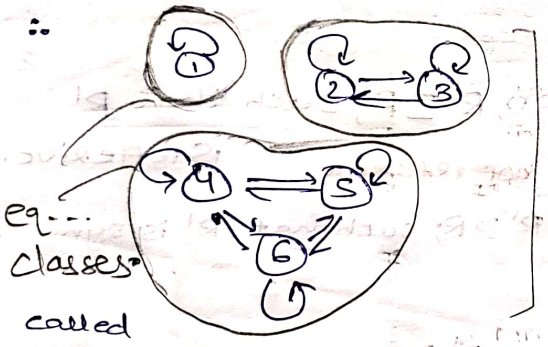
= E(2) or E(3)

= E(4) or E(5) or E(6)

* if $a \in Eq(x)$ & $a \in Eq(y)$ then $Eq(x) = Eq(y)$ (prove by symmetric, transitive).

∴ elements from two different equivalent classes (even though same equivalent relation) don't have a relation between them.
→ (a,b) doesn't belong to E.

∴

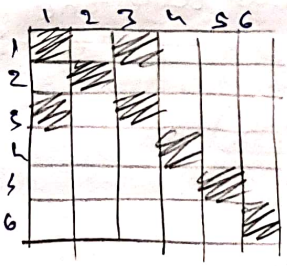


this is a equivalence relation

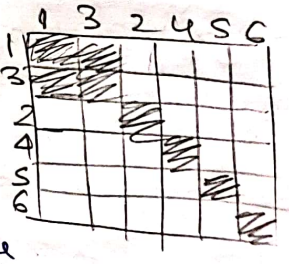
eq... classes called "cliques"

* An equivalence relation E is its own reflexive, symmetric, transitive closure.

- mind you;



is also an eq. class after sorting; acc. to equivalence class.



we get, square blocks along the diagonal.

* we can completely define a equivalence relation by assigning the elements of S; into any class each.

say $\{a, b, c, d\} = S$.

$\{a,b\}\{c\}\{d\}$ — set partitioning.

this partition has (a,a) (a,b) (b,a) (b,b) (c,c) (d,d) } eq° relation.

---

* lets see; how to introduce order among elements of a set:-

→ Landscape of transitive reln:-

irreflexive:	Strictly partially ordered sets. ≥	-transitive. -symm -irreflexive. Empty set.
reflexive:	partially ordered sets.	equivalence relation.
	Anti symm:	Symm:

∴ posets (partially ordered sets) are
- antisymmetric (∵ only one of a>b / b>a)
- transitive (∵ a>b, b>c means a>c for order existance)
- reflexive (a≥a)

↓

acyclic:

like this, cycles won't exist.
(∵ transitive & anti-symmetric).

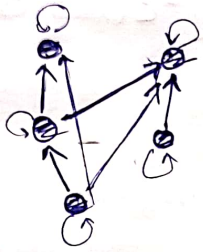
* "partially" ordered; because; not every 2 elements can be compared.

(if so, then called totally ordered).

i.e. there could be (a,b) s.t. $a \not\leq b$ & $b \not\leq a$ (i.e. order among a,b not def.)

↳ ≤ implies related.

* Poset (is a relation on S). can be drawn as.



- transitive
- antisymmetric (no $\leftrightarrows$ on any edge)
- reflexive (all have self loops)

poset; denoted as $(S, \preceq)$

* divisibility poset: $(S, |)$

for $a,b \in S$; if $a|b$ then $a \preceq b$.

is "related".
(i.e. (a,b) is element in relation)

- check reflx: $a|a$ ✓
- check transi: $a|b, b|c \therefore a|c$ ✓
- check antisymm:

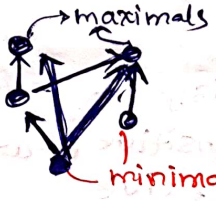
if $a|b \& b|a \Rightarrow a=b$ ✓✓
reflexive.

→ Extremals & extremum:-

* minimal: $x \in S$ & $\nexists y \in S-\{x\}$ s.t. $y \preceq x$, then $x \rightarrow$ minimal. (doesn't say that all $y \in S$; $y \geq x$ ∴ Some would be undefined)

maximal: $x \in S$ & $\nexists y \in S-\{x\}$ s.t. $x \preceq y$ then, $x \rightarrow$ maximal.

* Need not be unique.
* Need not exist. (in case of infinitely long S)
* every finite poset has atleast one maximal & one minimal. (Induction proof on $|S|$)



---

* Greatest: $x \in S$ if $\forall y \in S$, $y \preceq x$, then x is greatest.

Least: $x \in S$ if $\forall y \in S$, $x \preceq y$, then x is least.

either unique or D.N.E

even finite posets, may not have "greatest" elem.
∵ first of all; all $y \preceq x$ may not be present.

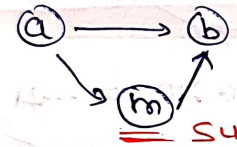
* Other relations from posets:-
  - reflexive reduction
  - transitive reduction

* $\prec$ is reflexive reduction of $\preceq$; iff $\preceq$ is reflexive closure of $\prec$ (smallest such reflexive rel^n.)
& $\prec$ is irreflexive relation.

Eg: $a \prec b$ if $a \neq b$ & $a \preceq b$.

* $\sqsubseteq$ is the transitive reduction of $\preceq$; iff $\preceq$ is transitive closure of $\sqsubseteq$ & $\forall a,b \quad a \sqsubseteq b \longrightarrow \nexists (m \notin \{a,b\}), a \preceq m \preceq b$

meaning:-

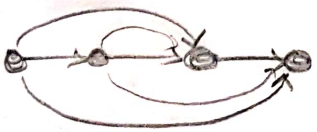


$\equiv$ such an "middle node, on path of $a \to b$"
$m \notin \{a,b\}$ shouldn't be there. if $a \sqsubseteq b$.

• well-defined for finite posets:-

$a \sqsubseteq b$ iff $a \preceq b$ & $\nexists m \notin \{a,b\}, a \preceq m \preceq b$.

(prove this by induction)



$a \preceq b$

transitive rel^n

(is not t. rel^n; ∵ its t. closure is not $\preceq$)

Like, drawing the skeleton

• need not exist for infinite sets: (eg; for $(\mathbb{R}, \leq)$)

there is a number, btw every two numbers!

$\sqsubseteq$ defined as such, would simply be the equality operator.

But! $\leq$ won't be its transitive closure.

* transitive reduction (of a transitive relation); if exists; is unique.

* divisibility poset $(\mathbb{Z}^+, |)$:

$\sqsubset$ is the transitive redn; where ($a \sqsubset b$ iff a/b is prime.)

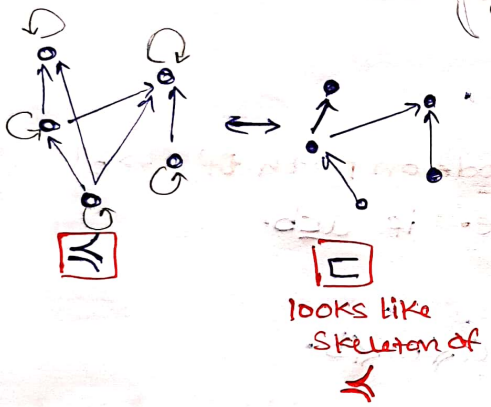
then

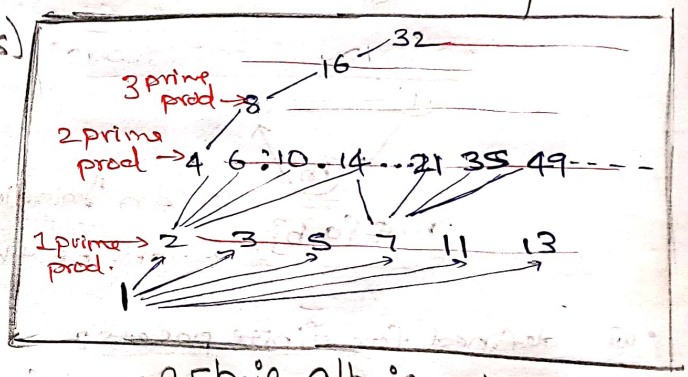
$\sqsubset$ is the t.reduct^n of $(\mathbb{Z}^+, |)$'s reflexive reduction.

Hasse diagram:-

* for a poset $(\preccurlyeq)$; the transitive reduction $(\sqsubset)$ of its reflexive reduction $(\prec)$ has all the information about poset $(\preccurlyeq)$

$\because$ $\preccurlyeq$ is ($=$ (self loops) + transitive closure of $\sqsubset$).





$a \sqsubset b$; if $a|b$ is prime.

* Bounding elem:-

poset $(S, \preccurlyeq)$ & $T \subseteq S$;

then

maximal in T ✓
minimal in T ✓
Greatest in T ✓
Least in T ✓

Upper bound for T:- $x \in S$; s.t. $\forall y \in T \quad y \preccurlyeq x$

Lower bound for T:- $x \in S$; s.t $\forall y \in T, \; x \preccurlyeq y$

- ~~Greatest~~ least upper bound:- Least in $\{x \mid x$ u.b. for $T\}$
- greatest lower bound:- greatest in $\{x \mid x$ l.b. for $T\}$

in terms of poset ordering....

→ Total order, Linear order:-

* In a poset $(S, \leq)$ every two elements are "comparable".

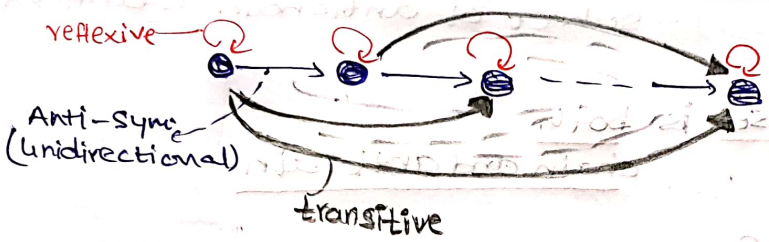
i.e. for $a, b \in S$

either $a \leq b$ or $b \leq a$;

(anti symm) if both are true; $\Rightarrow a = b$

• transitivity in poset def^n.

* can arrange all elements of S; in linear order with all possible edges pointing right (& all self loops).



→ Order extension:-

* A poset $P' = (S, \leq)$ is extension of $P = (S, \lesssim)$ if

$a, b \in S$; $a \lesssim b \rightarrow a \leq b$

every edge of P is there in P'.

* " Any poset can be extended to a total ordering"!

partially ordered set!
transitive!
antisymm! } essential for ordering.

(topological sorting)

(prove by induction on $|S|$)

Induct^n step:-

1) remove minimal element
2) extend the remaining to a total ordering.
3) reintroduce the minimal element as the least element.

→ Chains & antichains:-

in a poset $(S, \preceq)$

- $C \subseteq S$ is a chain;
  if $\forall a, b \in C$;
  $a \preceq b$ or $b \preceq a$
  i.e. total ordering
- subset of chain is chain.

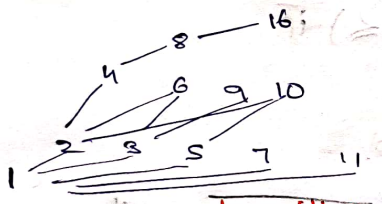
- $A \subseteq S$ is an antichain;
  if $\forall a, b \in A$
  neither $a \preceq b$ nor $b \preceq a$ unless $a = b$
  (relation not defined)
  only selfloops
  $(A, \preceq) \triangleq (A, =)$
- subset of antichain is antichain.

* Singleton set is both chain and antichain.

- Hence; for any chain C; antichain A;

$$|C \cap A| \leq 1.$$

*



transitive reduction. Hasse diag. easy chains

in divisibility poset $(\mathbb{Z}^+, |)$:

1 → 2 → 4 → 8 → 16 chain

2 3 5 7 11 antichain

* **Height:** (in a poset)

for an element a, in S;

- height(a) = length of longest chain with 'a' as maxima.
  poset $(S, \preceq)$
- at least, height(a) = 1; since $\{a\}$ is a chain.

* for $\mathbb{Z}^+$; if $n = p_1^{d_1} \cdot p_2^{d_2} \cdots p_n^{d_n}$

height $= 1 + \sum d_i$

∴ primes → height(2)

* $A_h = \{a \mid \text{height}(a) = h\}$

then A_h is antichain. (possibly empty).

(if not, then one element will have height $h+1$)

* Mirsky's theorem:-

least no. of antichains, needed to partion the elements of S is equal to max of height of element.

or

height of poset

or

length of maximum chain.

Prove by induction on length of max chain : anti : chain

• Dilworth's theorem:-

least no. of chains to partition elements of S is the size of largest antichain.

Proof: $|C \cap A| \leq 1$

& pigeonhole principle.vv.

form

we show answer is $\geq k+1$

& later show a solution $= k+1$

$\therefore k+1$ is least.

by induction, $= 1 + \{K\}$

↓

Antichain of all minimals

* prove something like; if $|S| = n$;

then $P(S, \preceq)$ poset

exists a chain of length $\geq \sqrt{n}$

or

anticlain of length $\geq \sqrt{n}$.

Functions :-

* $f: A \longrightarrow B$.

domain CODOMAIN · corresponding domain.

$B \neq$ range (or) $Im(A)$

- all elements have single image in B. for sure.

∘ composition ✓✓.

1) Surjection : (Onto).

- $f: A \longrightarrow B$.

$\forall y \in B \ \exists x \in A, \ f(x) = y$

- $Im_f(A) \equiv B$.

2) Injective : (one-One)

- $f: A \longrightarrow B$

if $f(x_1) = f(x_2) \longrightarrow x_1 = x_2$.

- function f is invertible

if $\exists \ g: B \longrightarrow A$ · such that

$gof = Id_A$.

- $Im_g(B) \equiv A$

- f need not be bijection; for it to be invertible.

∵ oneone implies existance of $g: B \longrightarrow A$ s.t.

$g(f(x)) = x$

[g need not be invertible]

3) Bijection :-

- $f: A \longrightarrow B$.

$f^{-1}: B \longrightarrow A$

$fof^{-1} = Id_B$

$f^{-1}of = Id_A$.

- codomain = range. (also foronto !)

* if gof is

one-one:-

then ↓

f is one one

(necessary, but not sufficient)

onto:-

then ↓

g is onto.

(necessary, but not sufficient).

6. Counting:-

- String of length 'k' from a alphabet of size 'n'.

  $num\_strings = (n)^k$

- Strings from binary {0,1}

  1011 | can be used to depict subsets of a set.
  0100 |

$P(n,r) = {}^nP_r$

$C(n,k) = \binom{n}{k} = {}^nC_k$

* $^nC_r + {}^nC_{r-1} = {}^{n+1}C_r$

- a recursive definition : base case: (written in blue)
- builds pascals ∆le

n \ k	0	1	2	3	4	5
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	

Base cases for recursive definitions.

→ conventions for n=0 or k=0:-

Think
english meaning of ^nC_k
"choose k from n"

∴ $^0C_0 = 1$

$^5C_7 = 0$ $^5C_0 = 1$ (empty string.√√.)

$^5C_{-1} = 0$

→ Bins and balls:-

	labelled balls	unlabelled balls.
labelled bins	function	multiset
unlabelled bins	set partition	integer partition

function:-

K balls; n bins

(i) unrestricted : $(n)^K$.

(ii) one one : ${}^nP_K$

(iii) onto:

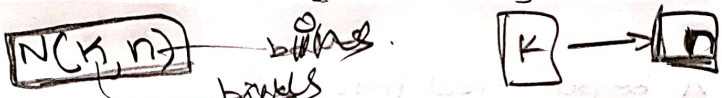
inclusion & exclusion

$$|R \cup S \cup T| = |R|+|S|+|T| - |R \cap S| - |S \cap T| - |T \cap R| + |R \cap S \cap T|$$

$$\text{num\_onto fn.} = \underbrace{n^K}_{\text{total}} - \{ \underbrace{{}^nC_1 \cdot (n-1)^K - {}^nC_2 (n-2)^K + {}^nC_3 (n-3)^K - \cdots}_{\text{opposite} \equiv \text{union of } 1^{st} \text{ elem no map, } 2^{nd} \text{ elem no map...}} \}$$

$$= n^K - {}^nC_1 (n-1)^K + {}^nC_2 (n-2)^K - {}^nC_3 (n-3)^K + \cdots$$

Multiset:-



meaning; an element can occur multiple times....

n balls; K bags:-
unlabelled

- multiset of size n, with elements as (K) labelled bags:-
- soln of $x_1 + x_2 + \cdots x_K = n$
  ↳ nonneg. int---

Stars & bars:-

n stars; K-1 bars.

no. of arrangements.

$$= {}^{n+K-1}C_{K-1}$$

Set partition:-

labelled balls into unlabelled bins:-

* $\{P_1, P_2, \ldots P_d\}$ is called a partition (set of sets) of A (set) if
  1) $\forall i \neq j,\ P_i \cap P_j = \phi$
  2) $P_1 \cup P_2 \cup \ldots P_d = A$
  3) $\forall i\ P_i \neq \phi$ · atleast 1 elem.

∴ $\{\{a,b\},\{c\}\} \triangleq \{\{c\},\{a,b\}\}$

* $S(k,n)$: sterling number of 2nd kind
  set of size 'k'; partitioned into exactly 'n' (partition sets; Bins; atleast 1 elem)

#ways A be partioned into atmost 'n' parts:
($|A|=k$)

(unconstrained version) $\sum_{m \in [n]} S(k,m)$

range $= \{1,2,3,\ldots n\}$

* $B_k = \sum_{m \in [k]} S(k,m)$ / all possible partitions
  (bell number)

---

what is $S(k,n)$ value?

• think the partitions are labelled;
then
count = onto functions, $k \to n$
$= N(k,n)$

labelled balls, unlabelled bins:

$$S(k,n) = \frac{N(k,n)}{n!}$$

(∵ each partition of size 'n' will be counted $n!$ times in numerator)

$N(k,n) = n^k - {}^nC_1(n-1)^k + {}^nC_2(n-2)^k - \ldots$

integer partition :-

K identical balls
n unlabelled bins.

= solutions to (count of no. of solutions).

$x_1 + x_2 + \cdots x_n = K$

$0 \le x_1 \le x_2 \le x_3 \le \cdots x_n$.

& integers...!

- if no bin empty :-

no. of such solutions are called partition number $P_n(K)$.

- no restrictions on bins :- (can be empty)

$= P_n(K+n)$

$P_n(K)$.
divide integer K into 'n' integers.
won't give closed form.
- recursively defined.

∵

$x_1 + x_2 + \cdots x_n = K;\ 0 \le x_1 \le x_2 \le \cdots x_n$

$\updownarrow$

$y_1 + y_2 + \cdots y_n = K+n;\ 1 \le y_1 \le y_2 \le \cdots y_n$

$y_i = x_i + 1$ bijection.

recursive definition of $P_n(K)$:-

Base cases :-

$P_0(0) = 1$; $P_0(K) = 0$; if $n > K$; $P_n(K) = 0$.

just for convenience... & also logical.

* $P_n(K) = \{(x_1, x_2, \cdots x_n) \mid x_1 + x_2 + \cdots x_n = K,\ 1 \le x_1 \le x_2 \le \cdots x_n\}$

non empty bins

$$P_n(K) = P_n(K-n) + P_{n-1}(K-1)$$

x_1 is not 1
so, now
$2 \le x_1 \le x_2 \le \cdots x_n$

x_1 is 1
so;
$1 \le x_2 \le x_3 \le \cdots x_n$

P.T.O.

n \ k	0	1	2	3	4	5	6	7	8
0	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1
2	0	0	1	1	2	2	3	3	4
3	0	0	0	1	1	2	3	4	5
4	0	0	0	0	1	1	2	3	5
5	0	0	0	0	0	1	1	2	3
6	0	0	0	0	0	0	1	1	2

$P_3(7) = P_2(6) + P_3(4)$

both are defined !!

→ base cases.

→ obtained recursively.

*no good closed form for $P_n(k)$.

Hence:-

	labeled balls (k)	unlabelled balls (k)
labelled boxes (n)	function: no restriction:- n^k 1-1: $P(n,k)$ onto: $N(k,n)$	multi-set no rest.: $C(n+k-1, k)$ 1-to-1: $C(n,k)$ onto: $C(k-1, n-1)$
unlabeled boxes (n)	set partition no restriction: $\sum_{m \in [n]} S(k,m)$ 1-1: 0 or 1 onto: $S(k,n)$	integer partition no restriction: $P_n(k+n)$ 1-1: 0 or 1 onto: $P_n(k)$

$P(n,k) = \frac{n!}{(n-k)!}$

$C(n,k) = \frac{n!}{(n-k)! \cdot k!}$

$N(k,n)$ = one one k to n = $n^k - {}^nC_1(n-1)^k + {}^nC_2(n-2)^k - \dots$

$S(k,n) = \frac{N(k,n)}{n!}$ ★

set partition.

$P_n(k)$ is; 'k' being written as 'n' positive integers. (recursive def^n; no closed form).

Graphs:-

- very efficient algorithms known for relevant graph problems.

  Eg: breadth/depth-first algo
  shortest path algorithm.

- But many other graph problems are known to be NP-hard.

  Eg: Traveling Salesperson Problem (TSP)
  visit all cities, by travelling least distance.

- Graph: (simple graph):

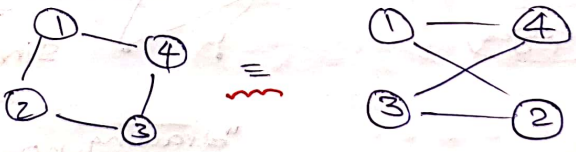
  $G = (V, E)$

  graph = (vertices, edges)

  V → nodes
  E → edges.

  where $E \subseteq \{\{a,b\} \mid a,b \in V\}$.
  subset.

"Drawing" is not part of graph. its our convenience.



1 — 2
2 — 3
3 — 4
4 — 1

~~automorphism~~
automorphism:-

→ A relation with: no-self loops
  & all existing edges are bidirectional.

  ≡ A graph.

  ∴ graph ≜ irreflexive, symmetric relation on (V)

  no mention of transitivity.

→ non-simple graphs:-

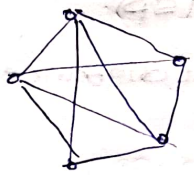
  Eg: - more than one edge b/w two nodes : multigraphs.
  - allow weights on edges (weighted graphs).

→ Terms:-

- Complete Graph K_n → if this is subset of G; clique in G.

  n nodes; with all possible edges b/w them.

  $E = \{\{a,b\} | a,b \in V\}$.



K_5

$^nC_2$ edges

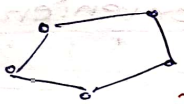
never say $G = K_n$

• G isomorphous to K_n

- Cyclic graph: C_n

  $V = \{v_1, v_2 -- v_n\}$

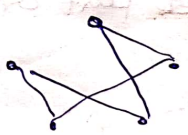
  $E = \{v_1 v_2\} \{v_2 v_3\} -- \{v_{n-1}, v_n\}$ & $\{v_1 v_n\}$.



C_5

n edges

note that:



also C_5

Since; i am able to go in loop.

"drawing, not part of graph. its own convenience".

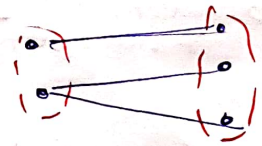
- Bipartite graphs:-

  $V = V_1 \cup V_2$; $V_1 \cap V_2 = \phi$

  & no edge in b/w two nodes of same "part"

  $E \subseteq \{\{a,b\} | a \in V_1, b \in V_2\}$

  subset.



* all C_n; n = even are bipartite

* k-regular graph:

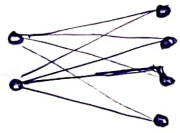
every node has k- edges.

- complete bipartite graph K_{n_1,n_2}:

$V = V_1 \cup V_2$

$V_1 \cap V_2 = \phi$

& $E = \{ \{a,b\} \mid a \in V_1, b \in V_2 \}$



$n_1 \cdot n_2$ edges

* later, hypercubes, trees.

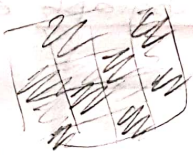
→ Graph isomorphism:-

dont see node values; compare structure only.

iso → same

morphosis → "to form"; "to shape".

∴ two graphs; (take it that their Nodes-set itself are different. $\{0,1,2,3,4\}$ & $\{a,b,c,d,e\}$).
if their shape is same; then isomorphism.



* Formally:-

$G_1 = (V_1, E_1)$ $G_2 = (V_2, E_2)$
are isomorphous;
if there exist bijection $f: V_1 \to V_2$

!

∴ $|V_1| = |V_2|$ (num_elements)

such that

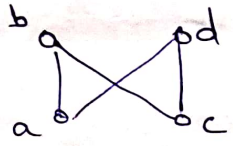
$\{a,b\} \in E_1 \iff \{f(a), f(b)\} \in E_2$

double implies.

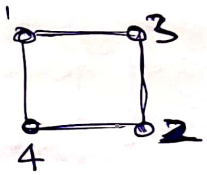
$f(E_1) \equiv E_2$

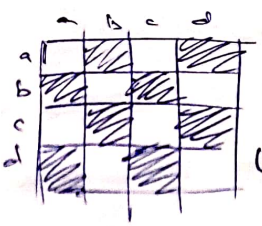
drawing may be different; but graph doesn't have a drawing element no.

Eg:

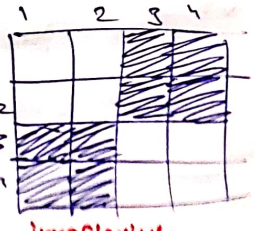


&





~~adjacency~~ adjacency matrices. (node type is irrelevant)



- irreflexive, symmetric.

- irreflexive, symmetric.

f:

a — 1
b — 3
c — 2
d — 4

is the needed bijection f.

* checking for isomorphism of two graphs
(two adjacency matrices)
is tough. no efficient algo. known.
(hashing + bruteforce) ✓ okay.

→ Invariants of isomorphism :-
(useful to predict the failure of isomorphism)

useful to prepare hash functions to quickly disprove isomorphism.

$|V_1| = |V_2|$ (∵ f is bijection)

$|E_1| = |E_2|$

degree sequence.
others exist;
but can't be used to show that $G_1 G_2$ = isomorphous

Can be used to show, $G_1 G_2$ are not isomorphous.

* χ - chromatic number.
min. no. of colours.

if not equal {invariant entity} for both G_1, G_2.

→ Subgraphs :-

Subgraph of $G = (V, E)$ is $G' = (V', E')$

$V' \subseteq V$ & $E' \subseteq E$.
(thats it!)

Steps :-
1) remove some elements. (nodes)
2) remove their edges.
3) remove some more edges. (still a subgraph)

• induced subgraph :-
dont do step-3:
just remove some node
& their associated edges.

→ Terms (this is what makes Graph theory – graph theory):-

- **walk** (of length K)
  (K edges transversed. could be 0.

  - a walk from node a to node b is a series of nodes

    $v_0\ v_1 \cdots v_K$ such that

    $v_0 = a$
    $v_K = b$

    & $\forall i \in \{0,1,\cdots K-1\}\ \{v_i, v_{i+1}\} \in E$.

  (there can be repetitions. if no repeating node → path).

  - length could be 0.

- **path:**

  if a walk has no repeating node, then its a path.

- **cycle:**

  A walk of length $K \geq 3$;

  a → b → a is not a cycle! but; its a walk. - not a path.

  C_K: { start node = end node;
  no other two nodes are same! }

  ∴ path with equal begin & end nodes.

  !

* A graph is acyclic; if it has no cycles.

trees ↔ acyclic graphs ??

i.e. C_K is not a subgraph of G. for any K.

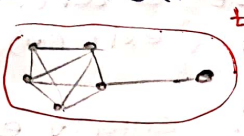
→ Connectivity:-

* whether there exists a path b/w two nodes.

if walk exists → path exists. (we trim down the walk).

Relation : Connected(u,v) is equivalence relation.

refle ✓
symm ✓
trans ✓



this complete is a connect class. dont see edges only lol.

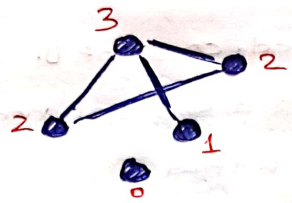
∴ equivalence classes give connected components of graph G.

2 components.

→ **Degree of a node :-**

for each node $v \in V$; degree = no. of edges incident on V.

(for simple graphs)



∴ number of edges $= \frac{1}{2} \cdot \sum_{v \in V} \deg(v)$

Degree sequence :-

sorted list of degrees.

$= \{0\,1\,2\,2\,3\}$

* degree sequence is **invariant** under **isomorphism**.

→ can be used to deny/confirm isomorph-- but not to "prove" isomorph- "necessary, but not sufficient!"

→ **Eulerian trial & circuit :-**

* **Eulerian trial :-** Bridges of Königsberg.

a walk visiting every edge exactly once. !

- if walk is a closed walk; start node = end node. eulerian circuit.

- **eulerian trial exists :-**

* there are (dunno converse) atmost 2- odd degree nodes.

Bcoz;

a trial is a walk.

- a start node
- a end node...

$\{v_0\ v_1\ v_2 \cdots v_k\}$ be the walk.

let $\text{enter}(v) = \{\{v_{i-1}, v_i\} \mid v_i = v\}$;

$\text{exit}(v) = \{\{v_i, v_{i+1}\} \mid v_i = v\}$;

∴ enter(v), exit(v) is a partition of edges incident on V.

if V is intermediate; $|\text{enter}(v)| = = |\text{exit}(v)|$

∴ degree = even. ✓

if start node ≠ end node;

then enter (end node) = ~~odd~~ 1 + exit (end node)

exit (start node) = 1 + enter (start node)

∴ Odd degree for both...

if start node = end node; | closed walk

then

enter = exit

∴ even degree. ✓

visiting every edge already implies visiting every node.

* only 1 node with odd degree can't happen;

bcoz;

$$\sum_{v \in V} \deg(v) = 2 \times |E|$$

even!

also a eulerian trial.

* Eulerian circuit:- A closed walk; visiting every edge exactly once.

∴ start node = end node.

eulerian circuit exists → no odd degree nodes.

*

What about converse?

if no odd degree nodes & all edges are in one connected component

then G must have eulerian circuit.

Proof: induction! but over what!?

(Say such epic 'proofs' base on induction!!.

or 'statements'

1) G has a cycle.

Why?(Becoz, say no cycles exist; & start a walk from some node. you can walk endlessly ⇒ ∞ nodes ↓ contradict^n).

2) remove cycle.

- still no odd degree node;

but now we have many connected classes.

3) Inductly; obtain E.circuit in each connected class.

4) Stich all these; onto the removed cycle, into a new, bigger circuit.

↳ if this exists; then drawing G would look nice.

→ Hamiltonian cycle:-

a cycle that contains all nodes in G.

a path with equal start, end node.

- No efficient algo; to check if a graph has hamiltonian cycle.

or condition;

like, for eulerian circuit, condition exists na.

- An NP-hard problem. Tough!

exp(n) growth.

not polynomial.

Recall: Checking isomorphism is hard;

but also believed to be not NP-hard.

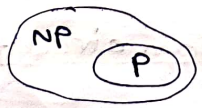
above check for hamiltonian cycle is NP.

P- NP problem:-

P problem: polynomial time complexity.

NP problem: exponential time complexity.

also; P ⊆ NP. (meaning; P doesn't have exp(n) growth)



But! we don't know whether its our inadequacy of writing a P-algo for a current-NP question;

or; there can't be written a P-algo as such.

→ Nobody yet "proved" that we "can't write" a P-algo. for NP-quest".

How to prove that! lol.

" if we solve one NP problem;

we solve all."

so we find different instances; where structure of problem is rich-enough to be NP. if we crack anyone; we are done.

→ Distance:-

* shortest walk between node u & node v is a path!
  - shortest walk is of great interest right!

Eg: Google maps. weighted edges.... dijkstra algo.
optic fibers as edges.
- probabilistic processes;
with shortest length ≡, most likely outcome.
↳ multiple "paths" exist ie.

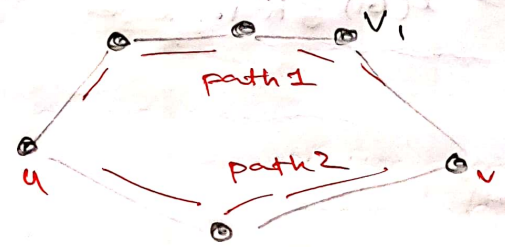
* length of shortest path = distance b/w u,v
no. of edges!
= ∞ if no path.
(i.e. belong to different connectivity class).

$$\min(w: u\text{-}v \text{ walk } |w|)$$

* diameter of G:
max. of all possible distance values.
or ≡ largest distance in a graph
can be ∞.

$$\max_{u,v} \text{distance}(u,v).$$

$$\equiv \max_{u,v} \min(w: \text{walk } u\text{-}v \text{ length}(w))$$



distance (u,v) = 2 ✓.

diameter = 3;
between u, v1.

Graph colouring:-

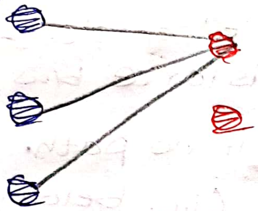
* a colouring is proper;
  if there is no edge between nodes of same colour.

  i.e. nodes of same colour form independent set.

But; not compulsory to have an edge b/w two nodes of diff. colour.
May or may not have edge.

Bipartite graph $\chi(G) \leq 2$

a 2-coloring is plainly visible



valid proper colouring.

* K-colouring:-
  a function (need not be onto.)
  $C: V \to \{1,2,3,\ldots K\}$
  s.t.
  $\forall x, y \in V$
  $\{x,y\} \in E \to c(x) \neq c(y)$
  not $\leftrightarrow$.

* trivial coloring:-
  N nodes → N colours.
  or more than N colours. (NUM if we don't use some. No where mentioned that ( ) got to be onto.)

* interesting:-
  min. no. of colours to colour the graph.

least no. of colours, in a proper coloring of G is chromatic number $\chi(G)$.

* G has a K-coloring $\Rightarrow$ $\chi(G) \leq K$ [upper]

  G has no K-1 coloring $\rightarrow$ $\chi(G) \geq K-1$. [lower]
  (maybe we get a contradict^n, if we have K-1 colouring attempt)

result:-

if H is subgraph of G

- if G has a K-coloring; so does H. (know that, some colours may not appear)
  need not be $\chi(G)$.
  any possible K. ✓

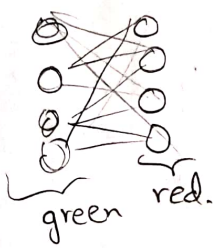
- $\chi(G) \geq \chi(H)$ [lower]

ROUGH:-

χ → isomorphism conserved.

$\chi(G) \leq |V|$

obvious na....



Colouring with 2 colours;

nodes with same color, no edge.

usually; K-colours;

no edge b/w same color nodes.

10 -

$V \rightarrow \{1,2,\dots K\}$

$\forall x,y \in V \quad \{x,y\} \in E \rightarrow C(x) \neq C(y)$

efficient algos; for coloring MANY SPECIAL kinds of graphs; with least colors.

→ In general; $\chi(G)$ is NP-hard.

Sir words "χ computation is rich-enough to be NP-hard".

$H \subseteq G$;

then

G has k-color → H has k-color

$\Rightarrow \chi(G) \geq \chi(H)$

lower bound on G.

$\Rightarrow$ G has K_n as a subgraph.

then $\chi(G) \geq n$.

G has C_{2K+1}

then $\chi(G) \geq 3$.

Chromatic number

$\chi(G)$.

least no. of colors....

G has k-coloring; $\Leftrightarrow \chi(G) \leq K$

G has no (K-1) coloring $\Leftrightarrow \chi(G) \geq K$.

Showing $K_{lower} \leq \chi(G) \leq K_{upper}$

1) show K_{upper} coloring for G

$\therefore \chi(G) \leq K_{upper}$

2) show $\chi(H) = K_{lower}$; where $H \in G$.

$\chi(G) = 2 \Rightarrow$ bipartite graph

Rough!

G is bipartite ⟷ no odd cycles. ∵ as subset. ∴ an odd cycle is NOT bipartite.

——→ proved by

$\chi(G) = 2$

$\therefore \chi(\text{subset}) \le 2$ $\quad \therefore \chi(C_{2k+1}) = 3$

$\therefore$ subset $\neq C_{2k+1}$

⟵ (no odd cycle)

(G is bipartite)

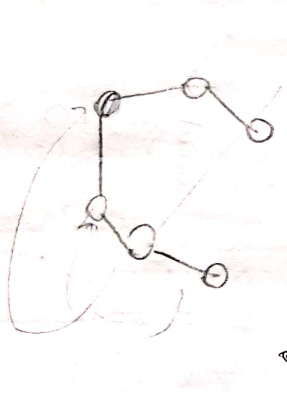
where G is not bipartite → atleast one odd cyc

↔ atleast one non-bipartite component.

↓

in this,

fix V.



$\forall x \in A$.

$A = \{x \mid dist(x,V) \text{ is even}\}$

$B = \{x \mid dist(x,V) \text{ is odd}\}$

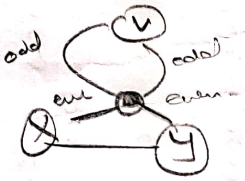
now;

A, B is not bipartite of connected

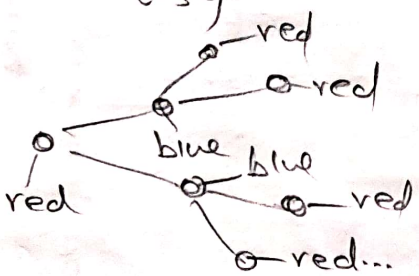
=>

$x, y \in A$

x — y exists. say



∴ for a bipartite graph; partition & 2-coloring is easy now....



i.e. fix node-v.

$A = \{x \mid dist(x,v) \text{ even}\}$

$B = \{x \mid dist(x,v) \text{ odd}\}$

} valid partition for any bi-partite graph. or 2-coloring graph

if this coloring comes wrong; then graph NOT bipartite

* Hence; if G has K_n as subgraph; $\chi(K_n) = n$

$\therefore \chi(G) \geq n$.

if G has C_n as subgraph; → $\chi(G) \geq 3$ in $n \equiv$ odd (C_{2k+1}'s $\chi() = 3$)

$\chi(G) \geq 2$ if $n =$ even. (C_{2k}'s $\chi() = 2$)

Hence; to give range for $\chi(G)$.

upper bound:- show a K_{up}-coloring;

$\chi(G) \leq K_{up}$.

lower bound:- shows a subgraph with $\chi(H) = K_{low}$

$\chi(G) \geq K_{lower}$.

* Isomorphism preserves coloring

=> preserves $\chi(G)$

→ complexity:-

efficient algos, for min-coloring special kinds of graphs.

- Bipartite

* But in general;

calculating $\chi(G)$ is NP hard. "its structure is rich enough, to be NP-hard!" - Mahesh Patil.

(if one NP hard is solved;
all NP-hard are cracked.

"They have the same structure"

So, we find as many NP-hard situations.
so that, one-day, a dumb-genius may crack any one of these;
& we all celebrate an end to the P-NP question)

→ Bipartite graph:-

we can easily have a 2-coloring in bipartite graph.

$\therefore \chi() \leq 2$.

(1) when no edges are there.

* C_{2n+1} is not bipartite. & $\chi(C_{2n+1}) = 3$

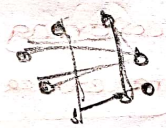
cycle

(Because, I can show 3-coloring & 2-coloring not possible

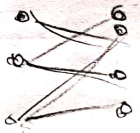
$\therefore \chi(G) = 3$).

→ Theorem:- to conclude if G is bipartite or not.

Even though we may draw G as



this COULD be a bipartite.



Drawing is not part of graph.

Bipartite ≡ no odd cycle.

no odd cycle ≡ $\chi(G) = 2$ or 1

* G, (with $|V| \geq 1$) is bipartite iff there exist no odd cycle as its subgraph.

(or)

* G is bipartite $\iff$ C_{2k+1} is not subgraph of G & thus $\chi(G) = 2$ or 1 okay! for any $k \in \{1, 2, 3, \dots\}$

Proof:

(G is bipartite → no odd cycle) is obv.

(if there is odd cycle; then $\chi(C_{2k+1}) = 3$ but parent graph $\chi = 2$ contradiction!).

* (no odd cycle → bipartite G)

we show; } equivalent.

(if G not bipartite → atleast one odd cycle)

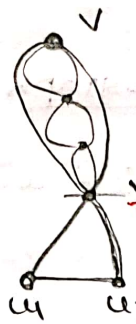
i) choose a vertex (V).

2) & partition the graph into $\{x : dist(x,V) \text{ even}\} = S_1$

$\{y : dist(y,V) \text{ odd}\} = S_2$

3) now; since not bipartite; atleast one element pair is connected in S_1 & S_2.

∴ let u_1, u_2 be such. with same parity of dist(u,v).



↳ least common vertex in the distance determining path

$V \to u_1$
$V \to u_2$
in this path; $V \to V'$ path got to be distance determining path to!

$\therefore |v'-u_1| = |u_1-v| - h$

$|v'-u_2| = |u_2-v| - h$

∴ the length of cycle: $v' \to u_1 \to u_2 \to v'$ is

$|u_1-v| + |u_2-v| - 2h + 1$

same parity. So sum is even

odd.

Hence we showed a odd cycle. when G is non-bipartite. (yay!).

---

corrects, followed from theorem:-

1) Hence if $\chi(G_1) > 2$; G_1 has atleast one odd cycle.
2) if $\chi(G_1) > 2$; G_1 is not bipartite.

** G has n nodes.

$\chi(G) = n \iff$ G is isomorphic to K_n.

proof:-

< G is isomorphous to $K_n \longrightarrow \chi(G) = n$ > obvious.

prov: < $\chi(G) = n \longrightarrow$ G is isomorphous to K_n >

contra positive approach — say G is not isomorphous to K_n.

$\Rightarrow \exists\, u,v \in V$ such that $\{u,v\} \notin E$

(n-coloring is okay! but not least possible)

1) then first colour with 'n' colours.
2) reduce usage of color(v) by making color(v) = color(u).

$\therefore \chi(G) < n$

→ Cliques and independent set:-

- Clique number $\omega(G)$:-

largest K, such that K_K is subgraph of G.

cliques are subgraphs of G; which are complete Graphs.

K_n.

trivial clique size = 1.

tough is: max clique size.

$= \omega(G)$.

$\therefore$ $\chi(G) \geq \omega(G)$

okay.

- Independence number $\alpha(G)$:-

largest K, such that G has a set of K nodes with no edges among them.

independent set.

- Nodes of a color form a independent set.

$\therefore$ atmost $\alpha(G)$ nodes can have same coloring.

but neednot.

$\Rightarrow$ at minimum; $\frac{n}{\alpha(G)}$ colours are needed for sure.

$\therefore$ $\chi(G) \geq \frac{n}{\alpha(G)}$

*

$\Delta(G)$: maximum degree, for a node in G.

Simply; $\chi(G) \leq \Delta(G)+1$

$\Delta(G)$ colours to sourrounding nodes

1 colour to the node in view.

$\therefore$ $\Delta(G)+1$ colours should suffice.

meaning;
we may have or may not. a colouring with less colours; but there exists a colouring with $1+\Delta(G)$ colours.

proof would be:-

1) induction on no. of nodes.
<build G from scratch>

2) induction step will be, to add a node x to the Graph $G-\{x\}$

3) induction says $G-\{x\}$ has $\Delta(G)+1$ coloring

$\because$ x will have atmost $\Delta(G)$ edges in G.

$\therefore$ there will always be atleast one color, which we can give to x, to maintain 'proper' coloring.

Problem models:-

Allocate "timeslot" for "courses", and avoid "conflicts".

colors — Nodes — edges.

min. coloring problem.

no edge should join nodes of same color.

no student should have 2-courses in same time.

conflict = common student for two courses

$\chi(G) = 1 \longrightarrow$ empty graph

$\chi(G) = 2 \longrightarrow$ bipartite graph. another def^n?

$\chi(G) = 3 \longrightarrow$ I don't give edit. (or) tripartite graph.

* if a graph is not empty; $\chi(G) \geq 2$

Examples:-

more graphs:-

K_n complete graph.

C_n cyclegraph.

K_{n_1,n_2}: complete bipartite graph.

Bipartitegraph.

1) path graphs P_n:

$V = \{1,2,\cdots n\}$

$E = \{\{i, i+1\} \forall i \in \{1,2,\cdots n-1\}\}$

$\chi() = 2$

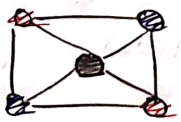
2) wheel graph:- W_n

$n \geq 3$

$V = \{hub\} \cup Z_n$

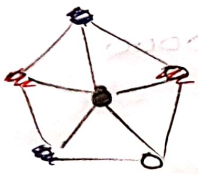
$E = \{\{hub, x\} | x \in Z_n\} \cup \{\{x, x+1\} | x \in Z_n\}$

W_4:



$\chi = 3$

W_5:



~~$\chi = 3$~~

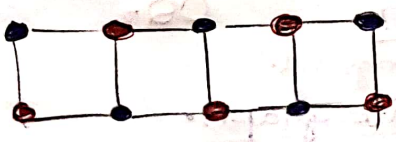
$\chi = 4$.

3) ladder graph:-

L_n (2n nodes) — heavy traps!!

$V = \{0,1\} \times \{1,2,3,\cdots n\}$

$E = \{\{(0,i),(1,i)\} | i \in [n]\} \cup \{\{(b,i),(b,i+1)\} | b \in \{0,1\}, i \in [n]\}$

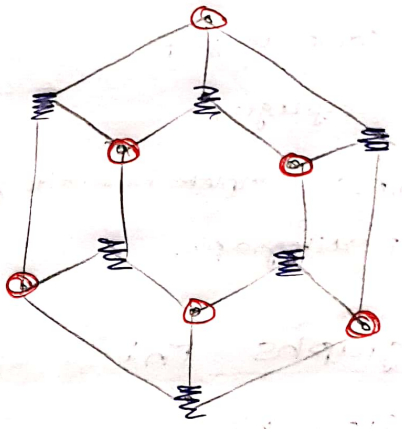


$\chi(G) = 2$

4) Circular ladder :- CL_n traps!

$2n$ nodes.

ladder, with ends joined.



if n = even, $\chi(G) = 2$

→ Hypercubes:-

Hyperellipsoid
Hyper Sphere
Hyperplane
Hypercube.
} in regards to higher dimensional beings.

Q_n Graph:-

nodes: all 'n'-bit strings. 00101 — 10101

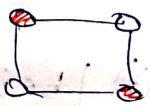
2^n in total

edges: x, y are edged; if they can be obtained by flip one ~~swap~~ of 0&1

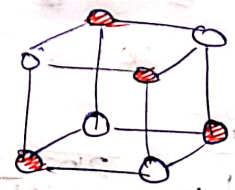
0000 — 0100 — 1100 — 1000

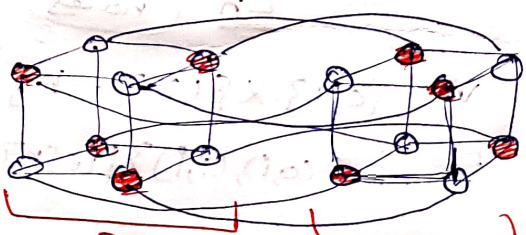
Q_0: .

Q_1: —

Q_2: 

trap!

Q_3: 

Q_4: 

$0Q_3$ $1Q_3$

* 2^n nodes. but diameter = n.
(n bit strings na.)

* Q_n is n-regular bipartite graph.
since, no odd cycle.

round trip has even swaps in total.
} think in terms of swapping of n-bits.

bipartite nature of Q_n:-

divide nodes into 2-groups.

even # of 1 in bit string: odd # 1 in bit strings.

→ Hence bipartite.

(Also, no odd loop. Logic in prev. page)

* Q_{n-1} is subgraph of Q_n.

→ Kneser Graph:-

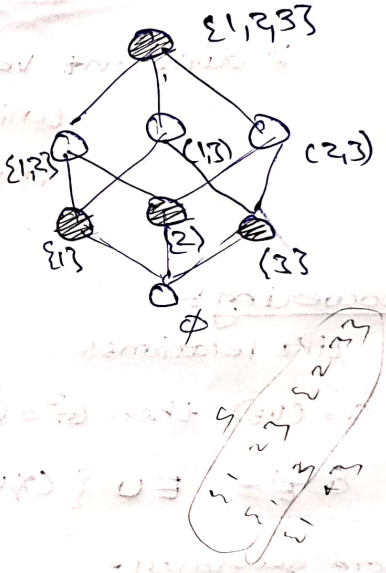
Q_n instead of bit strings;

1) nodes of Q_n represent subsets of $\{1,2,3,\cdots n\}$.

2) edges b/w u, v if

u ⊂ v or v ⊂ u.

& differ by single element.



* Now, → complement of graph.

$\overline{KG}_{n,K}$ is the set of nodes in K^{th} level (from bottom) in Q_n; ≡ subsets of $\{1,2,\cdots n\}$ of size K.

& edges:-

- present b/w subsets which have finite intersection.

* A clique in $\overline{KG}_{n,K}$ will mean, set of subsets; which intersect pairwise.

* erdos-ko-rado theorem:- if $K \le n/2$; then there exist, no larger cliques.

* Kneser Graph: $KG_{n,K}$ subsets of size K of $\{1,2,\cdots n\}$

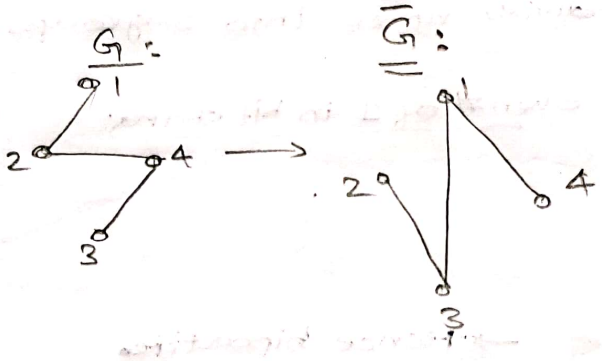
& edges b/w subsets; which are disjoint.

→ Graph operations :-

1) Complement:

$G_1 = V, E$

$\bar{G}_1 = V, \bar{E}$



2) union, intersection, difference, symmetric difference.

$G_1 = (V, E_1)$
$G_2 = (V, E_2)$
} same vertex set is a must.

if different vertex set;
union,
intersection
is still possible.

3) powering :-

Like relations :-

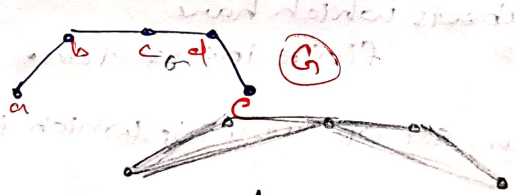
$G = (V,E)$ then $G^2 = (V,E')$

& $E' = E \cup \{ (x,y) \mid \exists w \ \{x,w\}\{w,y\} \in E \}$

* more generally;

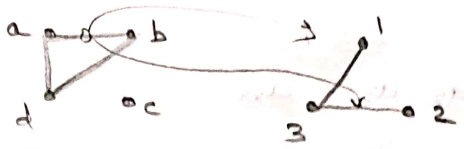
G^k has an edge (x,y) iff G has a path of length $<= K$ from x to y.

Eg:



(a,e) is in G^4
but not in
G^3, G^2, G.

imp:-



1) cross product:-

* $G_1 = (V_1, E_1)$ (cross of E_1, E_2)

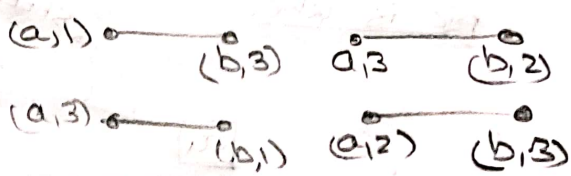
$G_2 = (V_2, E_2)$

then $G_1 \times G_2 = (V_1 \times V_2, E)$

where

Edge: $\{(u_1, u_2), (v_1, v_2)\} \in E$ iff $\{u_1, v_1\} \in E_1$ & $\{u_2, v_2\} \in E_2$

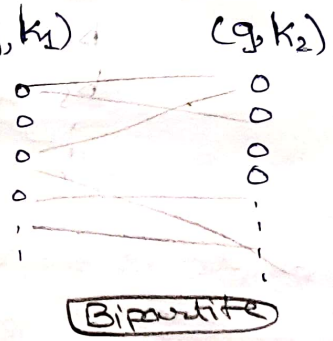
as if multiplying edges:-



{a,a} is not a valid edge.

Eg: $\langle G_1 \times K_2 \rangle$ is a bipartite graph.

any graph { say edges are k_1, k_2



2) Box product:-

$G_1 \square G_2 = (V_1 \times V_2, E)$

where

$\{(u_1, u_2), (v_1, v_2)\} \in E$ iff

$(\{u_1, v_1\} \in E_1$ and $u_2 = v_2)$

or

$(\{u_2, v_2\} \in E_2$ and $u_1 = v_1)$

Eg: $Q_m \square Q_n = Q_{m+n}$.

cross-product:- G_1:

$a_1\ a_2\ a_3 \cdots a_n$

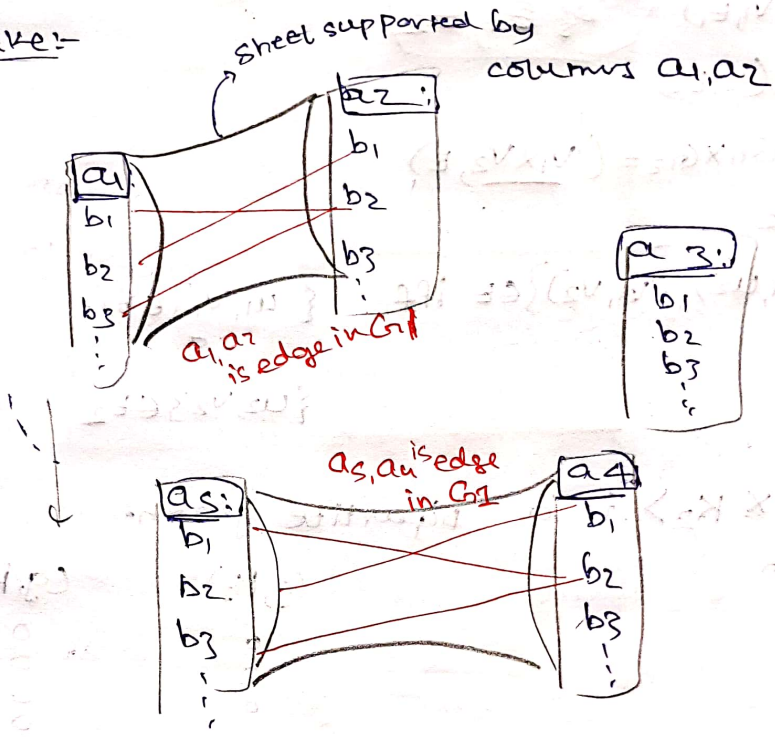
E_1

G_2:

$b_1\ b_2\ b_3 \cdots b_m$

E_2

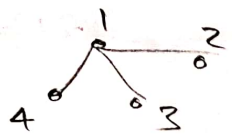
pic is like:-



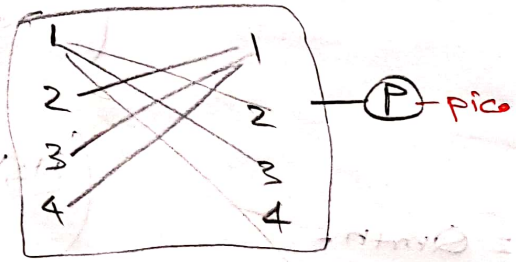
1) depth is added to graph G_1. Hence columns on a graph ground

2) first draw G_2 ka connectivity graph:- (bipartite)

if:



then:

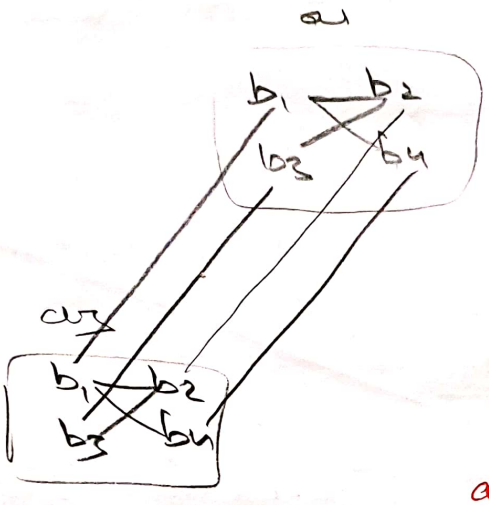


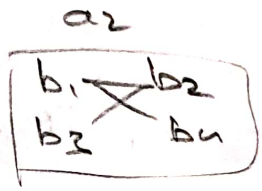
3) take this pic; & paste on a sheet supported by column of a_i, column of a_j, if a_j, a_i are edged.

Box product:

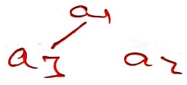
G_1:
a_1 a_2 a_3 ..

G_2:
b_1 b_2 b_3.

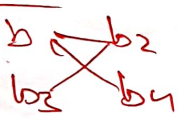




G_1:



G_2:



1) every node in G_1 has an image of G_2.

2) now; having each b_i in a level/layer, & draw G_1 in each layer.

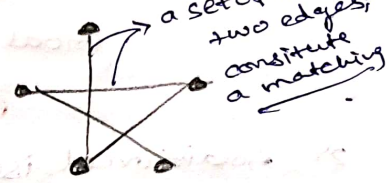
Graph matching:-

* Matching in a graph; is set of edges; s.t. no two edges have common vertex.

∴ every node gets "matched" with atmost one other node.

- a set $M \subseteq E$ such that $\forall e_1, e_2 \in M \; e_1 \cap e_2 = \phi$.

$G = (V, E)$



* A matching is a function, which mapps every node with atmost one other node.

Trivial matchings?

$M = \{\phi\}$

$M = \{e_i\}$ singleton.

Hard problem:-

Finding a maximum matching: matching of the largest size, for G.

- Not NP-hard.
- Efficient algos exist. <not discussed here>.

- maximal matching
- maximum matching
- perfect matching
- complete matching (for X; in $G(X, Y, E)$ Bipartite)

* perfect matching:-

All nodes are "matched" by M.

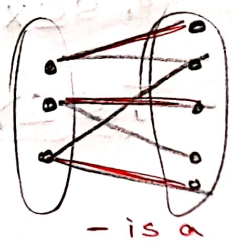
- may or may not exist.
- $|V|$ is even, if perfect matching exists.

→ Matchings in a bipartite graph:-

$G = (X, Y, E)$ bipartite.

* a complete matching from X to Y is is a matching such that $|M| = |X|$ every node in X is matched up.

necessary: $|Y| \geq |X|$.



- is a complete matching

→ practicality of matchings in bipartite graphs:-

X — one kind (entity) Y — other kind (resource).

1) Advertaisments & slots

each client, specifies their prefered slots.

Goal is to give every guy a slot.

2) Additional issues could be;

weights.

costs (minimum cost perfect matching)

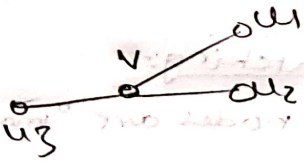
→ shrinking neighbourhood:-

* $G_1 = (V, E)$

for $v \in V$; we define v's neighbourhood as element.

(γ - small gamma; Γ → capital gamma)

$$\Gamma(v) = \{u : \{u, v\} \in E\}$$



$\Gamma(v) = \{u_1, u_2, u_3\}$

* for a set $S \subseteq V$

$$\Gamma(S) = \bigcup_{x \in S} \Gamma(x)$$

* for a bipartite; $G(X, Y, E)$ & $S \subseteq X$

$\Gamma(S) \subseteq Y$

* we say S is shrinking; if $|\Gamma(S)| < |S|$

* more generally for B in Y; S in X is shrinking, if

$|\Gamma(S) \cap B| < |S|$.

⇒ **Hall's Theorem:-** relates <Complete matching existance> with <Shrinking neighbour existance>.

" Bipartite $G = (X, Y, E)$ has a complete matching from X to Y iff no subset of X is shrinking."

necessary and sufficient.

() → () proving is easy.

() ← () proof: that if no subset of X is shrinking, then G has complete matching.

Claim:

→ proof by strong induction on $|X|$

Base case: $|X| = 1$ ✓

Induction step:-

- Suppose claim holds for all graphs with $|X| \leq K$.
- Given $G(X, Y, E)$ with $|X| = K+1$.

s.t. $\forall U \subseteq X, |\Gamma(U)| \geq U$.

1) pick $x \in X$ (arbitrary) & arbitrary neighbour y; $y \in \Gamma(x)$.

✓ case-1:-

* there exists a complete matching from $X - \{x\}$ to $Y - \{y\}$.

then, we are done ✓

2) case-2:-

* there dont exist a complete matching from $X - \{x\}$ to $Y - \{y\}$

By induction hypothesis;

$\exists S \subseteq X - \{x\}$ such that S is shrinking in $X - \{y\}$

(PTO)

But as per claim;

no subset of X is shrinking in Y.

$\Rightarrow$

S shrinking in $Y - \{y\}$.

S not shrinking in Y.

see lec. maybe

$\Rightarrow$ $y \in \Gamma(S)$

3) & $|\Gamma(S)| = |S|$!

& by induction hypothesis [if $|X| \leq k$ then complete Yes perfect matching if all subsets are non-shrinking

$\Rightarrow$ there exist a complete matching from

S to $\Gamma(S)$

actually, from S to Y.

but $\Gamma(S)$ is accurate.

4) showing: $\exists$ complete matching from $X - S$ to $Y - \Gamma(S)$:-

take $U \subseteq X - S$.

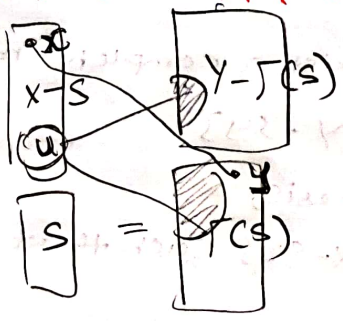
now; $U \cup S$ is a non shrinking subset in X.

$\therefore$ $|\Gamma(U \cup S)| \geq |U \cup S|$

$\Gamma(U) + \Gamma(S) \geq \Gamma(U \cup S)$ equals $|U| + |S|$ as $U \cap S = \phi$

$= |\Gamma(U) \cap (Y - \Gamma(S))| + |\Gamma(S)|$

$\therefore$



$\therefore \quad |\Gamma(U) \cap (Y - \Gamma(S))| \geq |U|$

$\therefore$ U is non-shrinking in $Y - \Gamma(S)$

By induct^{n} hypothesis -

$\therefore$ therefore, there exists complete matching from $X - S$ to $Y - \Gamma(S)$

$\therefore$ complete matching from X to Y.

when no shrinking subset.

claim proved.

Halls theorem.

Eg: the edge set of any bipartite graph in which all nodes have the same degree d, can be partioned into 'd' matchings.

steps:

1) prove 1 matching exists.

2) the remaining is d-1 regular bipartite.

(So recursion. or induction! words!

Vertex cover:-

- A vertex cover of a graph $G=(V,E)$ is a set C of vertices, such that every edge is covered by atleast one vertex in C.

  i.e. $C \subseteq V$ is a vertex cover, if $\forall e \in E$, $e \cap C \neq \phi$.

* Trivial vertex cover:-

- The set V.
- also the set $V-\{v\}$ for any node $v \in V$.

tough: finding the smallest vertex cover.

complexity:

It's NP-hard to find the smallest vertex cover.

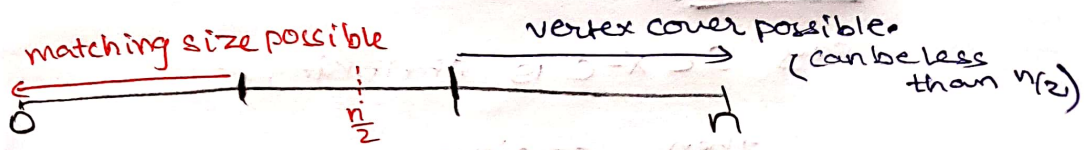
* Results connecting minimum vertex cover problem (NP) with max matchings problem (NOT NP hard)

1) In bipartite graphs, the size of smallest vertex cover = size of largest matching

2) In general graphs, they are within a factor of 2 of each other.

and,

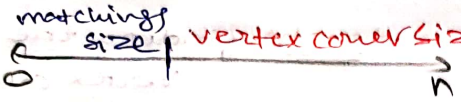
size of any vertex cover $\geq$ size of any matching.



Konig's theorem :-

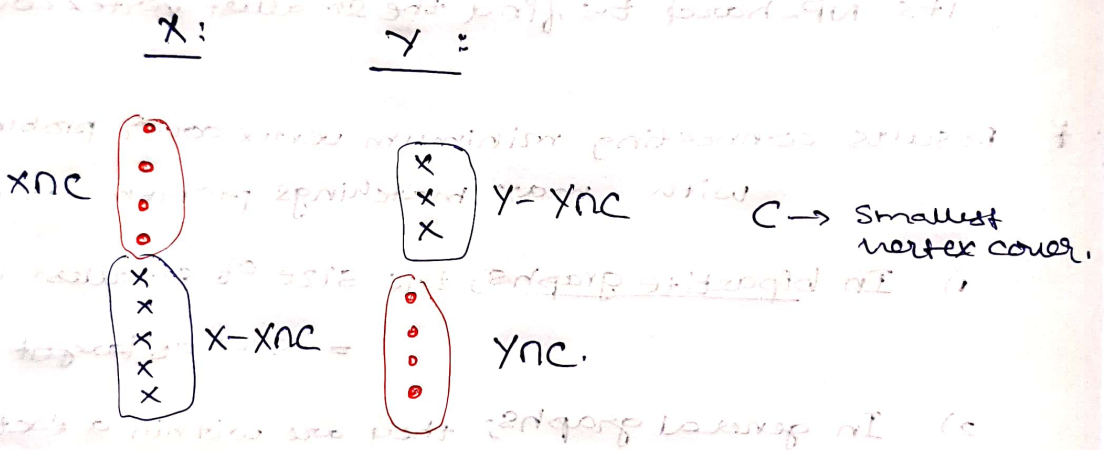
In Bipartite Graphs;

Size of largest matching = size of smallest vertex cover.



Proof: given a smallest vertex cover C; in $G(X, Y, E)$

we have matching M; $|M| \geq |C|$.



Claim: 1) we show, a complete matching exists from $X \cap C$ to $Y - Y \cap C$.

2) - By using Halls theorem :-

* we claim that there is no shrinking S in $X \cap C$.

otherwise :-

say $S \subseteq X - C$ is shrinking.

then consider 1) edges from S to $Y \cap C$
← covered by vertices in $Y \cap C$

2) edges from S to $Y - Y \cap C$:
← covered by vertices in $\Gamma(S) - (Y \cap C)$.

$\therefore (C - S) \cup [\Gamma(S) - (Y \cap C)]$ is a vertex cover smaller than C.

- contradict! So, no such shrinking S.

& this is smaller than S.

tough to find min. vertex cover

* easy to find max. matching → minimal vertex cover ?

easier to find maximal matching ⟹ cannot add one more edge to this matching, as it won't be a matching anymore.

M is maximal matching,
if ∄ e ∈ E−M
such that M∪{e} is also a matching.

way:

repeat until no further edge available:

1) choose arbitrary edge
2) delete other edges adjacent to this edge.

* if there exist a maximal matching of |M|.

then there exist a vertex cover of 2|M|.

(include both ends of edges in |M|)

∴

if C is smallest vertex cover,

& M is any maximal matching;

$$|M| \leq |C| \leq 2|M|$$

∴ can efficiently approximate the size of C.

within factor of 2.

---

* if I is independent set (no edges) in a graph G,

$\overline{I}$ got to be a valid vertex cover. (need not smallest)

(all edges of G)

∴ smallest v.c. size = n − largest independent set size.

$\alpha(G)$

vertex cover.

independent set.

Graphs — Trees:-

Tree: A connected acyclic graph.

Forest: An acyclic graph. ⟩ each connected component is a tree.

- a single tree is also a forest.

Subgraph of a tree = forest ⟩ may not be tree.

* **leaf:** node of degree 1.

properties:-

1) every tree with atleast 2 nodes, has atleast 2 nodes which are leafs.

2) Deleting a leaf from a tree G, results in a tree G'. (useful in induction on trees. on |V|.) forest but connected.

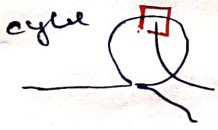
3) In a tree, only one path b/w two nodes u, v. (otherwise cycle exists.) but tree ≡ acyclic.

* In a tree, $|E| = |V| - 1$ — by induction.

* In a connected graph, if $|E| = |V| - 1$, then its a tree.

i.e. no cycle exists!

bcoz, if cycle exists;



remove one edge to make it acyclic. but connected!

then its a tree.

but $|E| = |V| - 2$ — contradict.

* Adding new edge to tree, makes it cyclic; with a single cycle.

So, no cycle in G if $|E| = |V| - 1$

means acyclic

* in a forest;

no. of trees = $|V| - |E|$

To connect these two;
Somehow make a poset
correspond to a bipartite graph

Dilworth's theorem:-

min. of chain partitions = length of longest antichain.

→ Min-Max results:-

in graph:-

* size of any matching ≤ size of any vertex cover.
* equality achieved in bipartite graphs. Konig's theorem

in posets:-

* size of any chain ≤ size of any antichain decomp.
* equality achieved Mirsky's theorem.

today:

* size of any antichain ≤ size of any chain decomp.
* equality achieved. Dilworth's theorem.

→ Dilworth's theorem:-

size of largest antichain = size of smallest chain decomp.

easy: ~~length~~ antichain ≤ length of chain decomp.

prove: there exists an antichain ≥ some. chain decomp.

---

* I thought somehow we make a bipartite G such that its vertex cover would imply a ~~cut~~ chain decomp. & its matching would imply a antichain.

spam

but no!

but something similar = comparision graph.

To connect these two; somehow make a poset correspond to a bipartite graph.

1) consider a poset $(S, \leq)$ with $|S| = n$.

2) construct a bipartite such that; $G(X, Y, E)$ (from S)

a vertex cover of size $\leq t$ in G $\Rightarrow$ antichain of size $\geq n-t$ in S

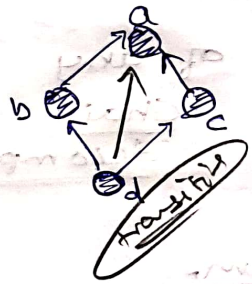
a matching of size $\geq t$ in G $\Rightarrow$ chain decomposition, size $\leq n-t$ of S

and then Konig's theorem gives an equal to both LHS..

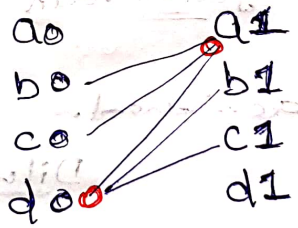
→ hence equal to both RHS.

3) let $G = (S \times \{0\}, S \times \{1\}, E)$ where $E = \{ \{(u,0),(v,1)\} \mid u \leq v \}$, $u \neq v$

poset:



then G_1 will be:-



C = vertex cover in G_1

$C = \{(a,1),(d,0)\}$

$B = \{ u \mid \exists b \in \{0,1\} \ (u,b) \in C \}$.

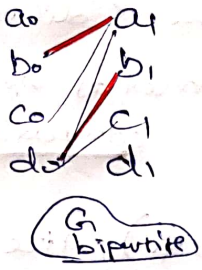
$\therefore B = \{a, d\}$

$A = S - B$

$A = \{b, c\}$ antichain!

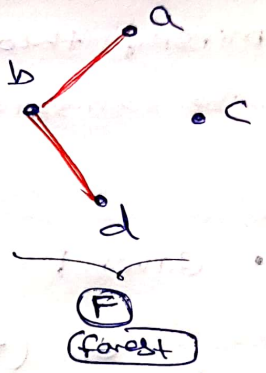
now; since $b_0 b_1 c_0 c_1$ are not in C, no edge in b/w these.
$\Rightarrow$ no rel^n among b, c in Poset
$\Rightarrow$ Antichain.

given matching M;



define graph

$F = (S, E^*)$



where

$E^* = \{ \{u,v\} \mid \{(u,0),(v,1)\} \in M \}$

* F is a forest.

with each connected component being a (path graph) (also a tree!)

* not called comparision graph! ↳ has the intermediate edges too!

(∵ for every node in F;

degree ≤ 2

one from (u,0) edge

one from (v,1) edge).

(& F has no cycles;

since cycle $v_0, v_1, \ldots v_k \Rightarrow v_0 \lesseqgtr v_1 \lesseqgtr v_2 \cdots \lesseqgtr v_k \lesseqgtr v_0$

or just reverse

).

& no. of components in F = $n - |M|$

∴ no. of chain decomp = $n - |M|$

→ comparision graph:-

* mirsky's theorem & dilworths theorem can be seen as statements on comparision graph of poset P.

- Given $P = (S, \leq)$, the comparision Graph is

$G = (S, E)$

where $E = \{ \{u, v\} \mid u \leq v, u \neq v \}$

- any induced subgraph of G is also a comparision graph.

! so we say comparision graph ⊆ perfect graph

chain = clique in G.

antichain decomp ≡ coloring in G

antichain = independent set in G (or) clique in $\bar{G}$.

chain decomp ≡ coloring in $\bar{G}$

mirsky's theorem:-

if G is comparision graph:-

$\omega(G) = \chi(G)$

Dilworth:-

if G is comparisin graph:-

$\omega(\bar{G}) = \chi(\bar{G})$

→ Perfect graph:-

G is perfect graph, if for every induced subset G';

$$\omega(G') = \chi(G')$$

usually; $\chi(G) \geq \omega(G)$

now; (=) "perfect".

then;

most brutal form of mirsky's dilworths.

- mirsky: every comparision graph is perfect
- Dilworth: the complement of comparision graph is perfect.

* fact: G is perfect $\iff$ $\bar{G}$ is perfect.

<perfect Graph theorem>

∴ ( ) states; Mirsky $\iff$ Dilworths.

* So. is K_n a perfect graph?

1) K_n is comparision graph for a total ordering with n element

2) mirsky's says that comparision graph is perfect.

3) Hence K_n is perfect. Woah!

* is complement of C_n, a perfect graph? $\forall n$.

we know C_{11} isn't perfect graph ($\omega = 2$, $\chi = 3$)

∴ $\bar{C_{11}}$ isn't perfect too! (perfect graph theorem)

Woah