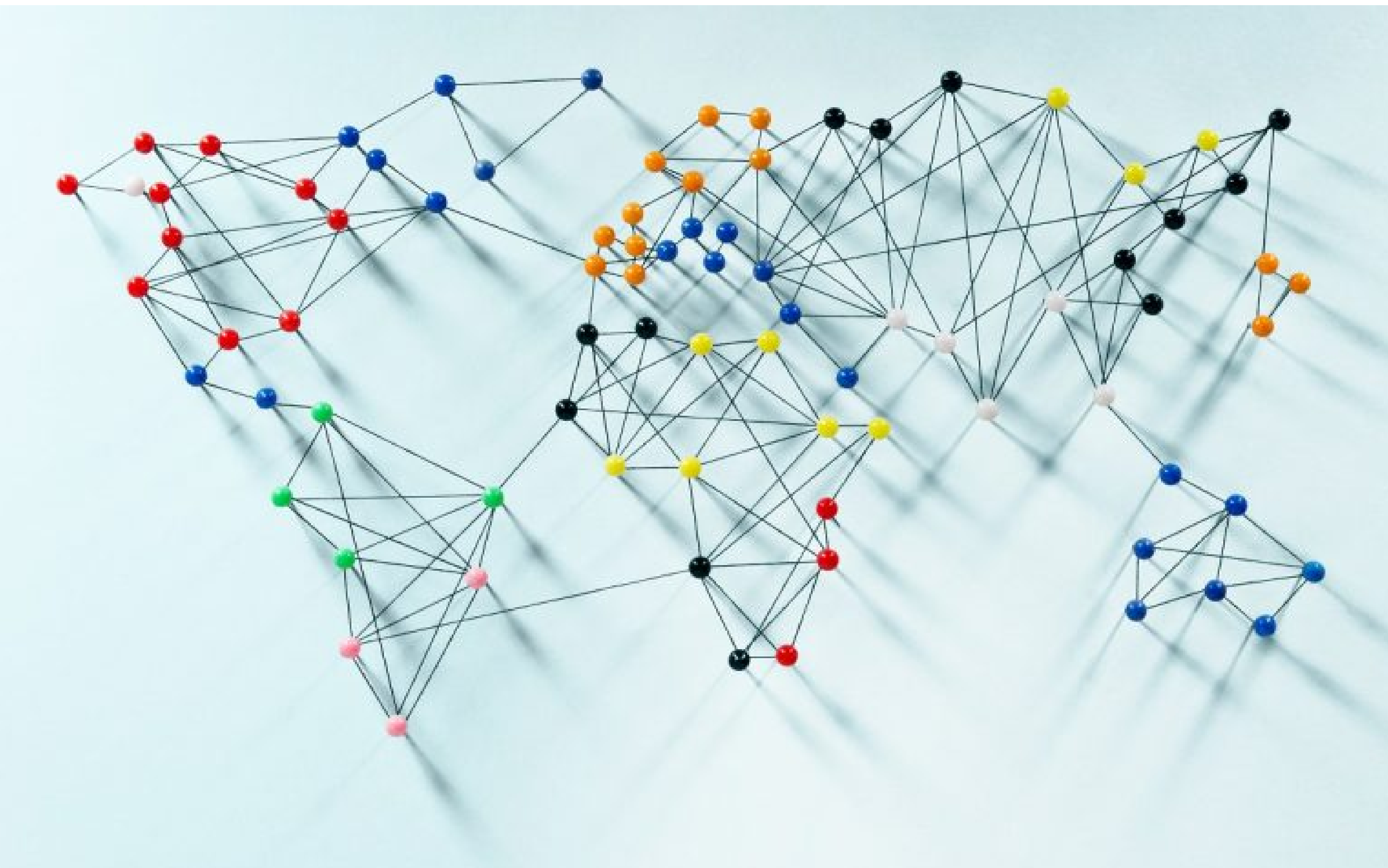
Graph Theory and Algorithms

An Introduction to the world of Graphs



Devansh Jain,
Indian Institute of Technology, Bombay

Contents

1 Preface **ii**

1 Preface

Preface

About the project

adedsd

About the report

sdsd

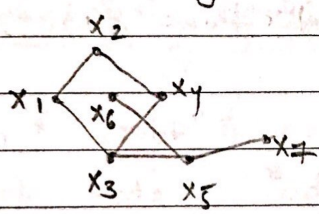
References and Acknowledgement

sdfsd

(23)

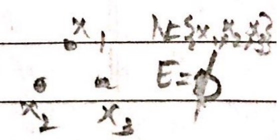
L-6 Graph are ~~very~~ incredibly useful structures in computer science. They come up in all sorts of applications, scheduling, optimization, communications, the design and analysis of algorithms.

⇒ Informally, a graph is just a bunch of dots and lines connecting the dots



⇒ A graph G is a pair of sets (V, E) where

- V is a non-empty set of items called vertices or nodes
- E is a set of 2-item subsets of V called edges. (can be empty set). ⇒



$V = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$

$E = \{ (x_1, x_2), (x_2, x_4), (x_4, x_3), (x_3, x_1)$
$(x_5, x_6), (x_3, x_5), (x_5, x_7) \}$

equivalent to $x_1 — x_2$

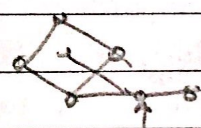
224

⇒ Two nodes x_i & x_j are adjacent if they're connected by an edge i.e. $x_i - x_j \in E$

⇒ An edge $e = (x_i, x_j)$ is said to be incident to its end-points x_i & x_j.

⇒ The number of edges incident to a node is called the degree of the node.

~~deg(x~~.



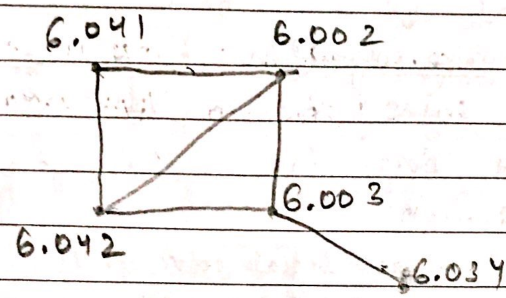
$deg(x_5) = 3$

⇒ Graph is simple if it has no loops or multiples edges. (multi-edge).

$|V| \rightarrow$ cardinality notation
i.e. no. of elements in set V.

Men-Women partners ratio

25



Slots
5-7
7-9
9-11
11-1
1-3

Graph-coloring problem

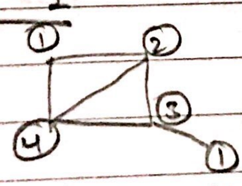
Given a graph G and K colors, assign a color to each node so adjacent nodes get different colors.

Minimum value of K for which such a coloring exists is the Chromatic Number of G

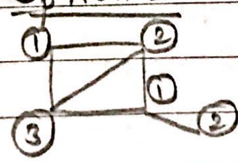
$\chi(G)$

→ Color ⇒ Slots
Nodes ⇒ Courses

Option 1



Option 2



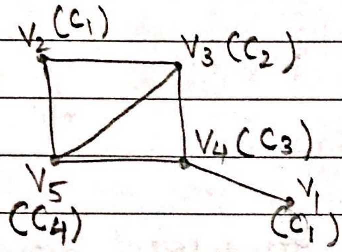
$\chi(G)=3$

2 colors can't work because of triangle formation.

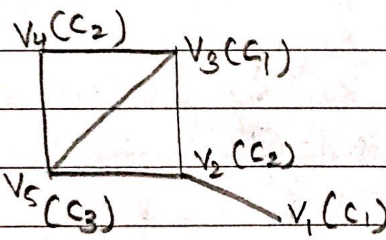
26

Basic graph coloring algorithm

1. Order the nodes $V_1, \ldots, V_n$
2. Order the colors $C_1, \ldots$
3. For $i = 1, 2, \ldots, n$
   Assign the lowest legal color to V_i



We know this is not best case.



So, if you change the ordering, you may get a better answer.

Now this algorithm is an example of what's known as greedy algorithm.
You just go one step after the next, taking the best you can do at each step. You never go back and try to make things better.

27

classmate
Date
Page

Thm. If every node in G has degree $\leq d$, then this basic algorithm uses atmost $d+1$ colors for G.

Proof By induction,

Induction Hypothesis:

// Never induct on d, induct on 'n' or 'e'
node

Predicate
$P(n) =$ " If every node in n-node graph G with max. degree d then this basic algorithm uses atmost $d+1$ colors for G."

Base case:

$n = 1 \Rightarrow |E| = 0 \Rightarrow d = 0 \Rightarrow$ one color

$\therefore P(1)$ is true

Ind. step: Assume $P(n)$ is true

Let G be any $n+1$ node graph.

Let d be the max. degree of nodes in G

Order the nodes: $V_1, \ldots, V_n, V_{n+1}$

Let G' be the graph after removing V_{n+1}

max. degree of nodes in $G \leq d$

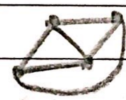
$\therefore$ Basic algorithm uses atmost $d+1$ colors for G'.

28

In G, color nodes $v_1 \ldots v_n$ same as G'.
v_{n+1} has atmost d neighbours therefore atleast one color is not present in v_{n+1}'s neighbours.
$\therefore$ Give v_{n+1} that color.

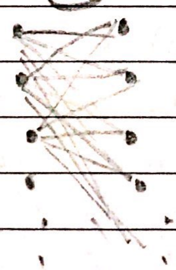
$\therefore$ $P(n+1)$ is true

$k_n \rightarrow$ n-node complete graph (or clique)



Degree $= n-1$
$\chi(K_n) = n$

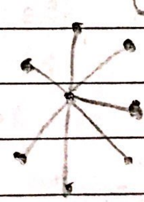
(n)



$d = n/2$
$\chi(G) = 2$

Bipartite graph

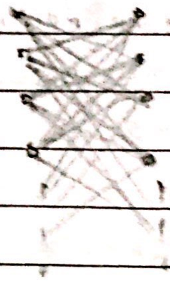
(n) Star graph



$d = n-1$
$\chi(G) = 2$

The Algorithm may be much better than the theorem

29



Good ordering (2 colors)

$(c_1)\ v_1$ • • $v_5\ (c_2)$
$(c_1)\ v_2$ • • $v_6\ (c_2)$
$(c_1)\ v_3$ • • $v_7\ (c_2)$
$(c_1)\ v_4$ • • $v_8\ (c_2)$
⋮

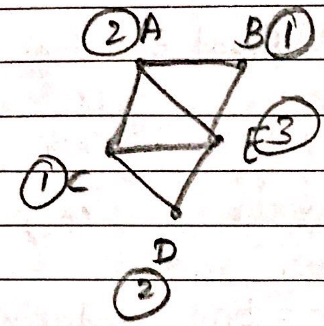
Bad ordering ($n/2$ colors)

$(c_1)\ v_1$ • • $v_2\ (c_1)$
$(c_2)\ v_3$ • • $v_4\ (c_2)$
$(c_3)\ v_5$ • • $v_6\ (c_3)$
$(c_4)\ v_7$ • • $v_8\ (c_5)$
⋮

A graph $G = (V, E)$ is said to be bipartite if V can be split into V_L, V_R so that all the edges connect V_L to V_R.

resource allocation problems (load balancing traffic on internet)

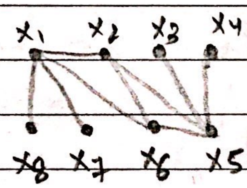
30



$\chi(G)=3$

Matching algorithm - usage: online dating agencies, assignments problems (matching interns to hospitals on Match day)

L-7 ⇒ Given a graph $G=(V,E)$, a matching is a subgraph of G where every node has degree 1.

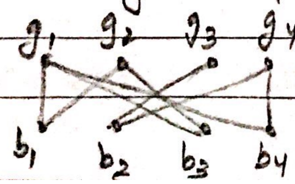


$\{x_1 - x_6, x_2 - x_5\}$ is a matching of size 2.

$\{x_1 - x_7, x_2 - x_6, x_3 - x_5\}$ is a matching of size 3.

Size 4 matching is not possible in this graph.

⇒ A matching is perfect if it has size $\frac{|V|}{2}$



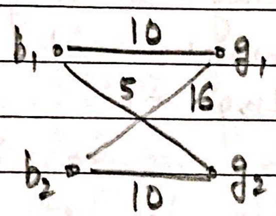
$\{g_1 - b_1, g_2 - b_3, g_3 - b_2, g_4 - b_4\}$ is a matching

31

Weights :

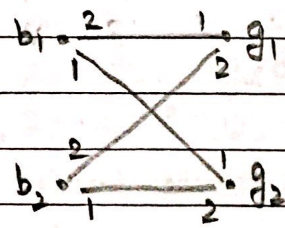
⇒ The weight of a matching (M) is the sum of the weights on the edge of M.

⇒ A minimum weight matching for graph G is a perfect matching for G with the minimum weight.



Min. weight matching
$\{b_1 - g_1, b_2 - g_2\}$
weight = 20

Preference list - similar to weights

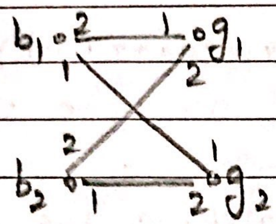


If $b_1 - g_1$, $b_2 - g_2$ then $b_1 - g_2$ will be rogue couple.

Given a matching M, x&y form rogue couple if they prefer each other to their mates in M.

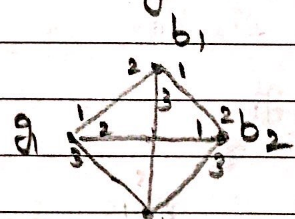
A matching is stable if there aren't any rogue couple.

32



$b_1 - g_2$ & $b_2 - g_1$ is stable matching

If b-g matching is only allowed then there always exist a stable perfect matching.
If b-b & g-g is allowed, it may not always possible to find stable perfect matching.



g_2 (Preference doesn't matter)

Theorem: There is no stable perfect matching

Proof: By contradiction.
If there is a stable matching, g_2 would be paired to someone
Without loss of generality (by symmetry) (the Δ is symmetric)
$\{g_2 - b_2 , g_1 - b_1\} \Rightarrow b_1, b_2$ form rougue couple.

33

classmate
Date ______
Page ______

1962
David Lloyd

Stable Marriage Algorithm (Gale - Shapley Algo)

- N boys & N girls
- each boy has his own ranked list of all the girls. Same for each girl
- to find stable perfect matching.

Eg:- 5 boys & 5 girls

1 ♂	CBEAD	A	35214
2 ♂	ABECD	B	52143
3 ♂	DCBAE	C	43512
4 ♂	ACDBE	D	12345
5 ♂	ABDEC	E	23415

Using greedy algorithm,
1 – C, 2 – A, 3 – D, 4 – B, 5 – E
Rogue couple : 4 – C

Using mating algorithm,
serenaders

1 ~~C~~BEAD
~~2 ABECD~~
3 DCBAE
4 ~~A~~CDBE
5 ABDEC

Girls	Day 1	Day 2	Day 3	Day 4	
A	2 4 5	5	5	5	A – 5
B		2	2 1	2	B – 2
C	1	1 4	4	4	C – 4
D	3	3	3	3	D – 3
E				1	E – 1

34

Need to show:

1. Algorithm terminates (Matching returned)
2. Everyone gets matched (Perfect matching)
3. No rogue couple (Stable matching)
4. It runs quickly
5. Fairness

Theorem 1 Algo terminates in less than N^2+1 days

Proof By contradiction.
Suppose, algo doesn't terminate in N^2+1 days.

Claim: If we don't terminate on a day, then atleast one girl had more than one boy and so she rejected atleast one boy who crosses her name.

If the algo doesn't terminate in N^2+1 days then atleast N^2+1 cross-outs have happened.

As there are N^2 names on N lists, there are atmost N^2 cross-outs. ($< N^2+1$)
Thus, not possible.

35

classmate
Date
Page

Predicate P = "If a girl G ever rejected a boy B, then G has a suitor who she prefers to B".

Lemma 1 : P is an invarient for the algo
Proof: ~~Base case:~~ By induction on number of days.
Base case: Day 0 (Vacuously true)
(No one rejected yet)
Ind step: P holds at the end of Day d.
Case 1 G rejects B on day $d+1$.
Then there was a better boy
$\Rightarrow$ P holds on Day $d+1$
Case 2 G rejected B before day $d+1$
P on day d $\Rightarrow$ G had a better boy on day d.
$\Rightarrow$ P holds on Day $d+1$.

Theorem 2 Everyone is married in Algo
Proof By contradiction
Assume that some boy B is not married,
then he was rejected by every girl
$\Rightarrow$ every girl by lemma 1 has a better suitor
$\Rightarrow$ every girl is married $\Rightarrow$ every boy is married
$\Rightarrow$ B is married.

35

classmate
Date
Page

Theorem 3 Algo produces a ~~stable~~ matching (No rogue couple)

Proof

Let B & G be any pair that are not married.

To prove: B & G aren't rogue.

Case 1 G rejected B
⇒ G has a better suitor than B (Lemma 1)
⇒ G married someone whom she prefers over B.
⇒ G is not rogue with B.

Case 2 G didn't reject B
⇒ B never serenaded G
⇒ G is lower in order than B's spouse.
⇒ B is not rogue with G

∴ No rogue couples.

37

classmate
Date
Page

Let S be the set of all stable matching
$S \neq \phi$ as algo produces one stable matching.

For each person P, we define the realm of possibility for P to be $\{Q \mid \exists M \in S \mid \{P, Q\} \in M\}$

A person's optimal mate is his/her favorite from the realm of possibility.

A person's pessimal mate is his/her least favorite from the realm of possibility.

Theorem 4 Algo marries every boy with his optimal mate.

Proof —

Theorem 5 Algo marries every girl with her pessimal mate.

Proof —

TMA → The Marriage Algorithm

38

classmate
Date
Page

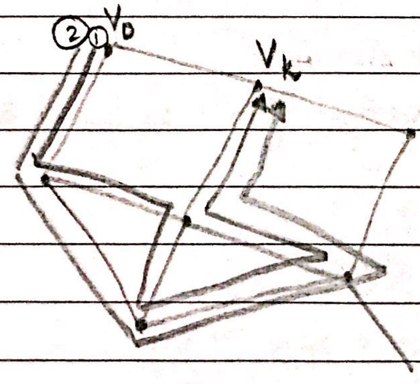
L-8
- Walks & Paths
- Connectivity
- Cycles & Closed walk
- Spanning Tree (ST)
- Min-weight spanning tree (MST)

Walks & Paths

⇒ A walk is a sequence of vertices that are connected by edges

Eg:- $v_0 — v_1 — \cdots — v_k$
Start end

It has k-edges ⇒ length = k



① → walk, not path
② → walk, and path

⇒ A path is a walk where all v_i's are distinct

39

Lemma 1 If $\exists$ walk from u to v, then $\exists$ path from u to v.

Proof $\exists$ walk u to v

By well-ordering principle: walk of minimal length.

$u = v_0 - v_1 - \cdots - v_k = v$

We prove that this walk (of minimal length) is a path.

Case $k = 0$ No edge length $= 0$

case $k = 1$ Just one edge length $= 1$

case $k \geq 2$

Suppose this walk is not a path

then $\exists_{i \neq j}$ $v_i = v_j$

$u = v_0 - \cdots - v_i = v_j - \cdots - v_k$

Now this a shorter walk

thus contradicting our assumption. ■

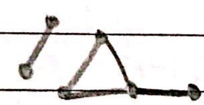
40

Connectivity

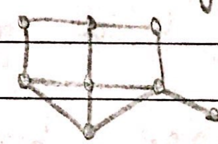
⇒ u and v are connected if there is a path from u to v.

⇒ A graph $G=(V, E)$ is connected if every pair of vertices $(v_1, v_2) \in V \times V$ are connected.

Not connected graph



Connected graph



Cycles & Closed walks

(~~also known as tours~~)

⇒ A closed walk is a walk starts and ends at exactly same vertex.

$$v_0 - v_1 - \cdots - v_k = v_0$$

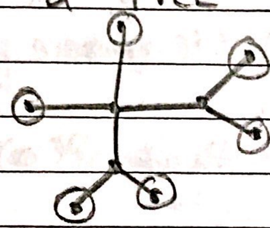
⇒ If a closed walk with $k \geq 3$ and if all $v_0, \ldots, v_{k-1}$ are distinct then it is called a cycle

41

Trees

⇒ A connected and acyclic graph is called a tree.

⇒ A leaf is a node with degree = 1 in a tree



Lemma 2 Any connected subgraph of a tree is a tree.

Proof By contradiction,
if the connected subgraph is not a tree then it must have a cycle.
As the subgraph is a part of a graph, the graph must have a cycle.
But the graph is a tree.
We get a contradiction. ∎

42

Lemma 3 A tree with n vertices has $n-1$ edges.

Proof By Induction

~~P(n)~~ Ind. Hypo. : P(n) = "There are $n-1$ edges in an n-vertex tree"

Base case: P(1) is true.

Ind. step: Assume that P(n) is true

Let T be an $(n+1)$-vertex tree.
Let v be a leaf of the tree.
Delete v; this creates a connected subgraph which is also a tree (Lemma 2)
By P(n): this subgraph has $n-1$ edges.
Re-attach v ∴ T has $(n-1)+1 = n$ edges (degree of $v = 1$)

P(n+1) is true

43

Spanning Trees

⇒ A spanning tree (ST) of a connected graph ~~G = (V, E)~~ is a subgraph ~~that~~ is a tree with same vertices as ~~the~~ graph.

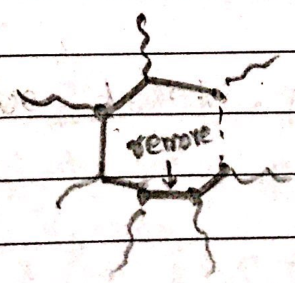
Theorem Every connected graph has a ST.

Proof By contradiction.

Assume a connected graph G with no ST.

let T be a connected subgraph of G and has same vertices as G and has minimum number of edges.

As T is not a spanning tree, it must have atleast one cycle.



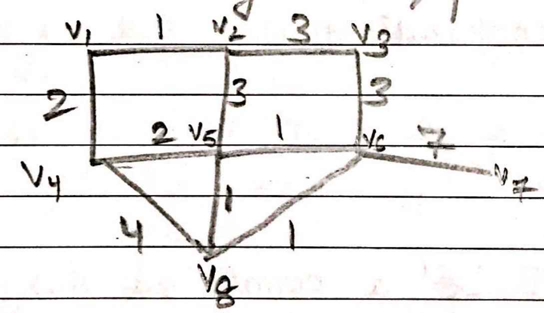
if we remove one edge of the cycle, we still have a connected subgraph.

44

Subgraph we obtain is smaller than T (w.r.t. number of edges) thus contradicting our minimum condition on T.

■

Minimum-weighted spanning tree



$ST_1 = \{ v_1 - v_4, v_4 - v_5, v_5 - v_2, v_5 - v_6, v_6 - v_3, v_6 - v_7, v_5 - v_8 \}$

Weight of $ST_1 = 2 + 2 + 3 + 1 + 3 + 7 + 1 = 19$

$ST_2 = \{ v_1 - v_2, v_1 - v_4, v_4 - v_5, v_5 - v_6, v_6 - v_3, v_6 - v_7, v_5 - v_8 \}$

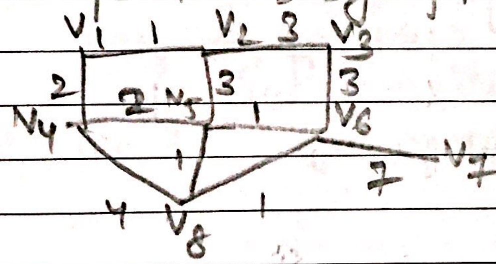
Weight of $ST_2 = 1 + 2 + 2 + 1 + 3 + 7 + 1 = 17$

45

⇒ The minimum spanning tree (MST) of an edge-weighted graph is defined as the ST of G with the smallest possible sum of edge-weights.

Algo

- Grow a subgraph one edge at a time, such that at each step:
  - Add the minimum weighted edge keeps the subgraph acyclic.



+1 ① $V_6 - V_8$

+1 ② $V_1 - V_2$

+1 ③ $V_5 - V_6$ ⇒ $V_5 - V_8$ ✗

+2 ④ $V_1 - V_4$

+2 ⑤ $V_4 - V_5$ ⇒ $V_4 - V_8$ ✗, $V_2 - V_5$ ✗

+3 ⑥ $V_2 - V_3$ ⇒ $V_3 - V_6$ ✗

+7 ⑦ $V_6 - V_7$

17

classmate
Date
Page

(46)

Theorem ~~For any connected weighted graph G, the~~ algo produces a MST.

Lemma: Let S consists of first k-edges by the algo, then $\exists$ MST $T=(V, E)$ for G such that $S \subseteq E$

Proof: (Assumed lemma as true)
Graph $G=(V, E)$
$|V|=n$

1) Suppose $< n-1$ edges are picked, then $\exists$ edge in $E-S$ that can be added without creating a cycle.

2) Once $n-1$ edges are picked, we know S is an MST.

Proof: By Induction:
$P(m) = \forall G \; \forall S$ consisting of first m selected edges $\exists$ MST of G $T=(V, E)$ such that $S \subseteq E$.

Base Case: $m=0 \Rightarrow S=\phi \subseteq E$ for any MST $T=(V, E)$

Ind. Step: Assume $P(m)$ holds
Let e denote the $(m+1)$th selected edge.
Let S denote the first m selected edges.

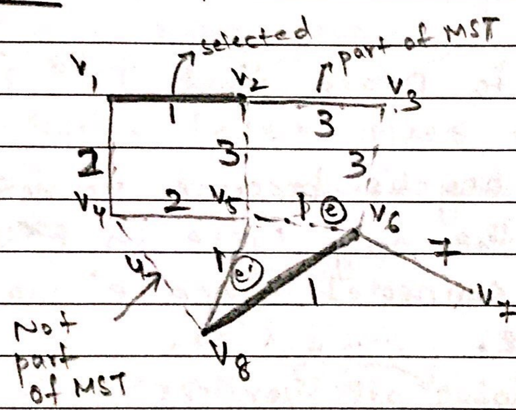
47

Let $T^* = (V, E^*)$ be MST of G such that $S \subseteq E^*$

Case 1: $e \in E^* \Rightarrow S \cup \{e\} \subseteq E^* \Rightarrow P(m+1)$ holds

Case 2 $e \notin E^*$

$S = \{v_1 - v_2, v_6 - v_8\}$



$E^* = \{v_1 - v_2, v_2 - v_3, v_1 - v_4, v_4 - v_5, v_5 - v_8, v_8 - v_6, v_6 - v_7\}$

$e = v_5 - v_6$

~~From the algo,~~

Algo $\Rightarrow S \cup \{e\}$ has no cycle
(This is how we choose $(m+1)^{th}$ edge)

T^* is a tree $\Rightarrow (V, E^* \cup \{e\})$ must contain a cycle. (Proof in book)
$=(V, E^*)$

this cycle has edge $e' \in E^* - S$

48

classmate
Date
Page

Algo could have selected e or e'
$\Rightarrow$ weight of $e \leq$ weight of e'.

Swap e and e' in T^*:

Let $T^{**} = (V, E^{**})$: $E^{**} = (E^* - \{e'\}) \cup \{e\}$

We need to prove that T^{**} is an MST.

- T^{**} is acyclic because we removed e' from the only cycle in $E^* \cup \{e\}$.
- T^{**} is connected since e' was on a cycle.
- T^{**} contains all vertices of G

$\Rightarrow T^{**}$ is ST of G.

weight of T^{**} = weighted sum of E^{**}
= Weighted sum of E^* − weight of e' + weight of e
= weight ~~sum of~~ of T^* − weight of e' + weight of e

As weight of $e \leq$ weight of e'
weight of $T^{**} \leq$ weight of T^*
And as T^* is MST, equality holds & T^{**} is also MST. $\square$

49

L-9 → L-10

Communication Networks

- great application of graph theory
- how do you route packets through networks

$N \times N$ Networks ($N = 2^k$).

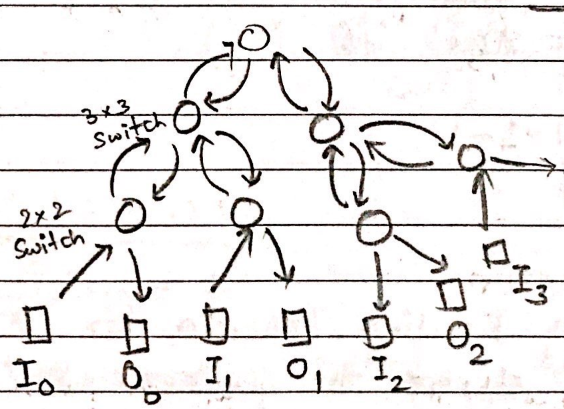
Complete Binary Tree

○ = switch (direct packets through a network)
□ = terminal (source and destination of data)

4 x 4 Network

Diameter = 6

Switch count = 7



$\pi(i) = i$
↳ Congestion = 1

$\pi(i) = N-1-i$
↳ Congestion = 4

Max Congestion = 4

$N \times N$ Network

Diameter $= 2(1 + \log_2 N)$

Max. Switch size $= 3 \times 3$

Switch count $= 2N - 1$

Max Congestion $= N$

50

Latency: Time that is required for a packet to travel from an input to an output.

Diameter of a network: Length of the shortest path b/w Input and Output that are farthest apart (Maximum length of shortest path

A permutation is a function $\pi : \{0 \dots N-1\} \to \{0 \dots N-1\}$ such that no two numbers are mapped to the same value
($\pi(i) = \pi(j)$ iff $i=j$)

Eg:- $\pi(i) = N-1-i$
$\pi(i) = i$

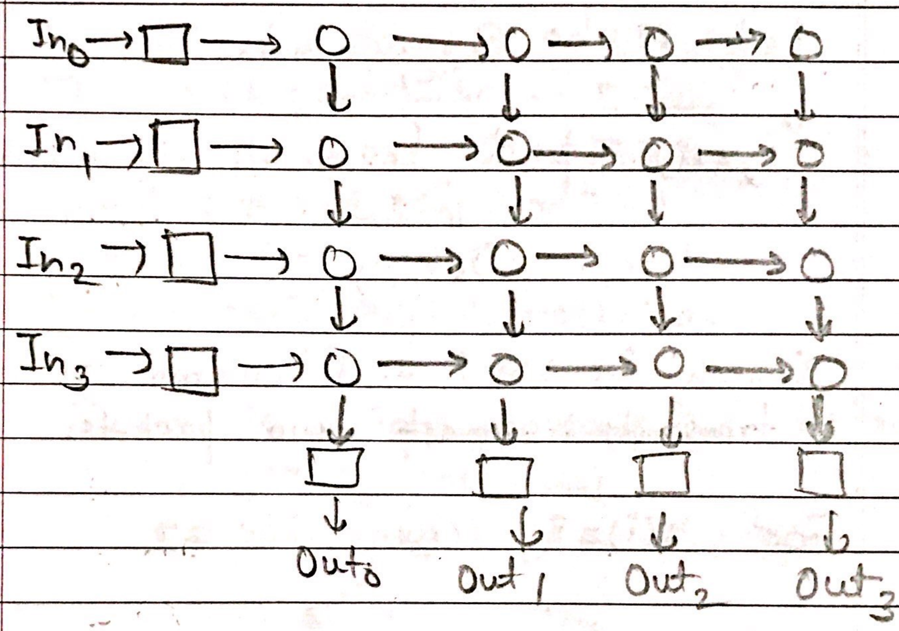
Permutation Routing Problem for π
- For each i direct the packet at I_i to $O_{\pi(i)}$
  path taken is denoted by $P_{i,\pi(i)}$

The congestion of paths $P_{0,\pi(0)}, \dots, P_{N-1,\pi(N-1)}$ is equal to the largest number of paths that pass through a single switch.

Max congestion = max possible value of congestion. (over permutations with best path)

51

2D Arrays (Grid structure)



4x4 Network	NxN Network
Diameter = 8	Diameter = 2N
Max. switch size = 2x2	Max. switch size = 2x2
No. of Switches = 16	No. of switches = N^2
Max. congestion = 2	Max. congestion = 2

52

classmate
Date
Page

Theorem ~~N·(A)*~~ Congestion of an N-input array is 2.

Proof Let π be a permutation.

$P_{i,\pi(i)}$ = path from In_i rightward to colum $\pi(i)$ and downward to $Out_{\pi(i)}$.

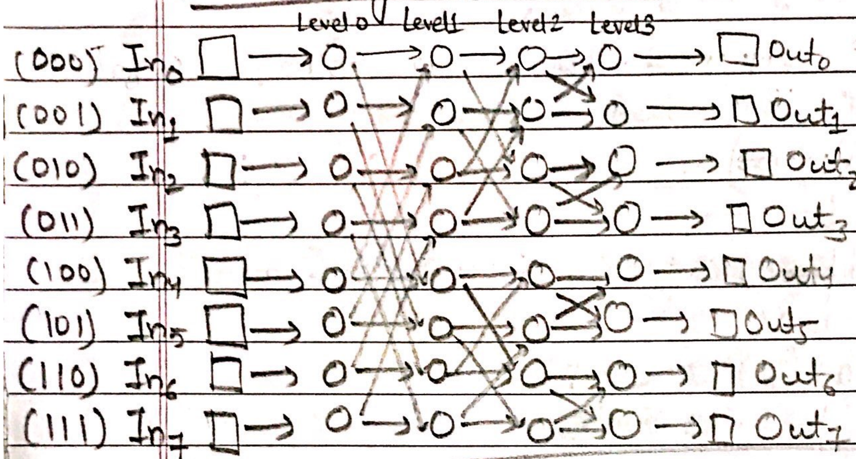
Switch in row i and column $\pi(i)$ ~~2 packets~~ transmits atmost two packets.

For $\pi(i)=i$, congestion = 2

$\boxed{\Xi}$

53

Butterfly Network ~~Structure~~



Switch is uniquely identified its row & col

$$\Big(\big(\underbrace{b_1 \cdots b_{\log N}}_{\text{binary rep. of input}}\big), \overset{\text{level}}{\ell}\Big)$$

$$\big((b_1 \cdots \overline{b_{\ell+1}} \cdots b_{\log N}), \ell+1\big) \qquad \big((b_1 \cdots b_{\ell+1} \cdots b_{\log N}), \ell+1\big)$$

complement of $(\ell+1)^{th}$ bit.

Routing

$$((x_1 \cdots x_{\log N}), 0) \longrightarrow ((y_1 x_2 \cdots x_{\log N}), 1)$$

$$((y_1 y_2 x_3 \cdots x_{\log N}), 2)$$

$$((y_1 \cdots y_\ell x_{\ell+1} \cdots x_{\log N}), \ell)$$

$$((y_1 \cdots y_{\log N}), \log N)$$

54

In3 $\longrightarrow$ Out5

Eg:- $011 \longrightarrow 101$

$$\begin{array}{c} \text{In3} \rightarrow (011, 0) \\ \downarrow \\ (111, 1) \\ \downarrow \\ (101, 2) \\ \downarrow \\ \text{Out5} \leftarrow (101, 3) \end{array}$$

N×N Network

Diameter $= 2 + \log N$

Max. Switch size $= 2 \times 2$

No. of Switches $= N(1 + \log N)$

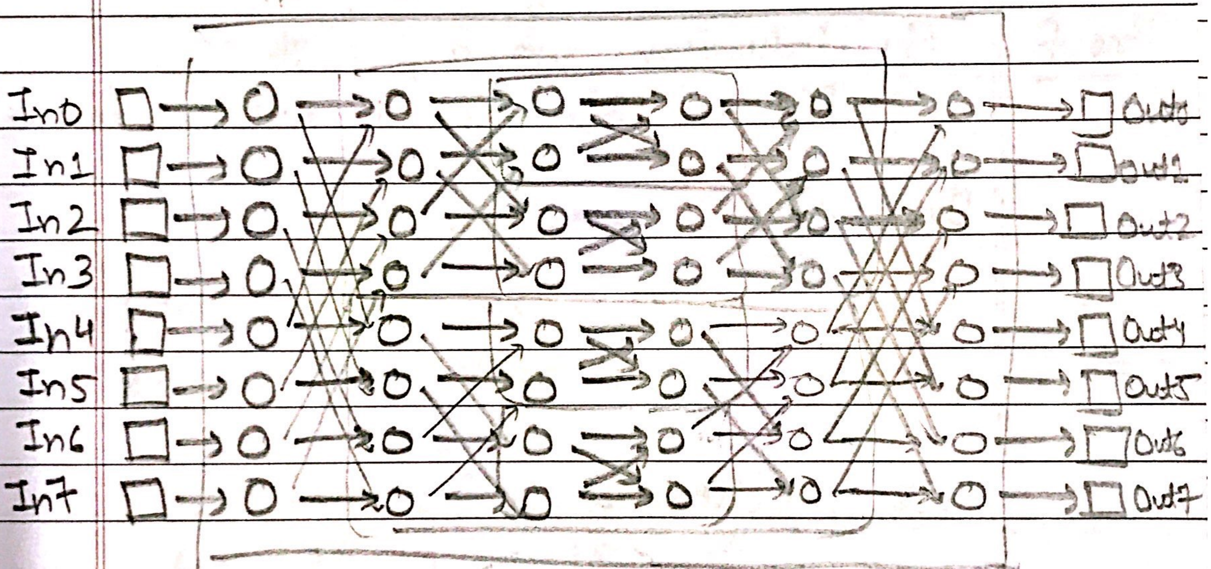
Congestion $= \sqrt{N} \quad (N \in 2^{2n})$

$= \sqrt{N/2} \quad (N \in 2^{2n+1})$

55

Benes

1960s - Benes, a Bell Labs researcher



N x N Network

Diameter = 1 + 2 log N

Max switch size = 2x2

No of switches = 2 N log N

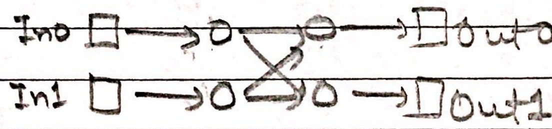
Congestion = 1

56

Theorem The Congestion of N-input Benes network is 1, when $N = 2^a$ for some $a \geq 1$

Proof By induction on a,
$P(a)$ = "This theorem is true for a".

Base case: $a = 1$ $N = 2$



$\pi(0) = 0$	$\pi(0) = 1$
$\pi(1) = 1$	$\pi(1) = 0$
Congestion = 1	Congestion = 1
(Just straight)	(Just cross-over)

Ind. Step: Assume $P(a)$ is true

Example: $\pi(0) = 1$ $\pi(4) = 3$
$\pi(1) = 5$ $\pi(5) = 6$
$\pi(2) = 4$ $\pi(6) = 0$
$\pi(3) = 7$ $\pi(7) = 2$

Constraint graph
If two packets must pass through different subnetworks then there is an edge b/w them.

57

L-10 Euler tours

L-9 Directed graphs

- Definitions
- # walks No. of
- strong connectivity
- DAGs

Tournament Graphs

Euler tour

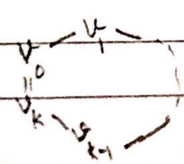
- he lived in Konigsberg
- Seven bridges (birth of graph theory)

An Euler tour is a walk that traverses every edge exactly once and starts and finishes at same vertex.

Theorem A connected graph has an Euler tour iff every vertex has even degree.

Proof $(\Rightarrow)$

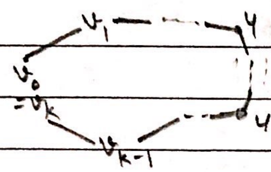
Assume $G = (V, E)$ has an Euler tour



58

classmate
Date
Page

Since every edge in E is traversed once:
clearly $\deg(u) = (\#$ times u appears in this tour $v_0 \dots v_k) \times 2$



$(\Leftarrow)$

For $G=(V,E)$, assume $\deg(v)$ is even $\forall v \in V$

Let $W: v_0 - v_1 - \cdots\cdots - v_k$ be the longest walk that traverses no edge more than once.

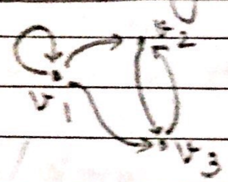
① $v_k - u$ not in $W \Rightarrow v_0 - v_1 - \cdots - v_k - u$ is longer than $W \Rightarrow$ Contradicts
$\therefore$ All the edges in E incident of v_k are traversed in W.

② $v_k \neq v_0 \Rightarrow v_k$ has odd degree in walk W
As we have showed that all edges in E of v_k are in W, so $\deg(v_k)$ is odd in G
$\Rightarrow$ Contradicts as $\deg(v)$ is even $\forall v \in V$.

59

classmate Date Page

Directed graph (digraphs)



indegree $(v_2) = 2$
outdegree $(v_2) = 1$

Theorem Let $G = (V, E)$ be an n-node graph with $V = \{v_1, \ldots, v_n\}$. Let $A = \{a_{ij}\}$ denote the adjacency matrix for G that is

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \to v_j \text{ is an edge.} \\ 0 & \text{otherwise} \end{cases}$$

Let $p_{ij}^{(k)}$ = # directed walks of length k from v_i to v_j.

then $A^k = \{p_{ij}^{(k)}\}$

$$A = \begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \qquad A^2 = \begin{bmatrix} 1 & 2 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 1 & 3 & 3 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

60

Proof $a_{ij}^{(k)}$ denote the $(i,j)^{th}$ entry in A^k

By induction:

Predicate → $P(k) =$ "Theorem is true for k"

$= \text{"} \forall_{i,j} \ a_{ij}^{(k)} = p_{ij}^{(k)} \text{"}$

Base case $k=1$

Edge $v_i \to v_j$: $p_{ij}^{(1)} = 1 = a_{ij}^{(1)}$

No edge $v_i \to v_j$: $p_{ij}^{(1)} = 0 = a_{ij}^{(1)}$

$\therefore \ \forall_{i,j} \ a_{ij}^{(1)} = p_{ij}^{(1)}$

Ind. step

Assume $P(k)$ holds.

$$p_{ij}^{(k+1)} = \sum_{h:\, v_h \to v_j \text{ is an edge in } E} p_{ih}^{(k)} = \sum_{h=1}^{n} p_{ih}^{(k)} \cdot a_{hj}$$

$v_i \overset{k}{\rightsquigarrow} v_h \xrightarrow{1} v_j$

$$= \sum_{h=1}^{n} a_{ih}^{(k)} \cdot a_{hj}$$

Ind. step

$$= a_{ij}^{(k+1)} \Rightarrow P(k+1) \text{ holds}$$

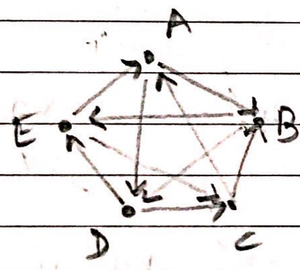
Matrix multiplication

61

⇒ A digraph $G=(V,E)$ is called strongly connected if $\forall\, u,v \in V$ $\exists$ directed path from u to v in G.

⇒ A digraph is called directed acyclic graph (DAG) if it does not contain any directed cycles.

Tournament Graph



either u beats v : $u \to v$
or v beats u : $v \to u$

⟶ E beats A, C and loses to B, D.

⇒ A directed Hamiltonian path is a directed walk that visits every vertex exactly once.

Theorem: Every tournament graph contains a directed Hamiltonian path.

62

classmate
Date
Page

Proof By induction (on number of nodes)

$P(n) =$ "Every tournament graph on n-nodes contains a directed Ham. path."

Base case : $P(1)$ holds

Ind. step : Assume $P(n)$ holds

Consider a tournament graph on $(n+1)$-nodes

Take out a node v. We are left with a tournament graph on n-nodes.

By $P(n)$, $v_1 \to v_2 \to \cdots \to v_n$

Approach ① If $v \to v_1$ is an edge, then $P(n+1)$ holds.
(if $v_1 \to v$ and $v \to v_n$) then
Claim: $\exists i$ such that $v_i \to v$ & $v \to v_{i+1}$
Proof by contradiction
If no such i exists and we know $v_1 \to v$ then $v_i \to v$ $\forall i$ (by induction)
But $v \to v_n$ contradicts this.

~~There is also proof by strong induction.~~
See it.

63

Approach ②

Case 1 $v \to v_1$

Case 2 $v_1 \to v$

Smallest i such that $v \to v_i$

$\Rightarrow v_k \to v \quad \forall \; k < i$

$\Rightarrow v_{i-1} \to v$

$v_1 \to \cdots\cdots \to v_{i-1} \to v_i \to \cdots \to v_n$

v

A new directed Hamiltonian path is form

$\therefore P(n+1)$ holds

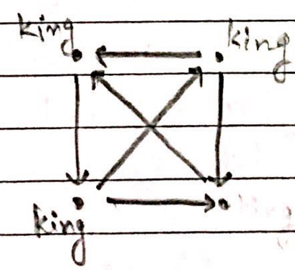
64

King Chicken Problem

either chicken u pecks chicken v : $u \to v$
or chicken v pecks chicken u : $v \to u$

u "virtually" pecks v if
- $u \to v$, or
- $\exists_w$ $u \to w \to v$

A chicken that virtually pecks every other chicken is called king chicken.



Theorem The chicken with highest outdegree is definitely a king.

Proof By contradiction,

Let u have highest outdegree.
Suppose u is not king

65

classmate
Date
Page

$\Rightarrow \exists_v : v \longrightarrow u$, and $\forall_w : u \to w \Rightarrow v \to w$

(outdegree of v) $\geq$ (outdegree of u) + 1

So, our assumption is false. ∎

L-11
- Relations
  - properties
- Equivalent Relations
- Partial Orders
  - Hasse diagram
  - Total order
  - Topological sort
- Parallel Task Scheduling
  - Dilworth's Lemma

Relations

$\Rightarrow$ A relation from a set A to a set B is a subset $R \subseteq A \times B$

Ex :- $R = \{(a,b) :$ student a is taking class b $\}$

$(a,b) \in R$: aRb , $a \not R\, b$

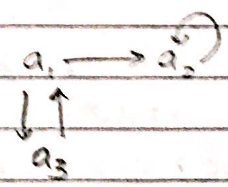
Relational Symbol

66

⇒ Relation on A is a subset $R \subseteq A \times A$

Eg:- $A = \mathbb{Z}$: xRy iff $x \equiv y \pmod 5$
$A = \mathbb{N}$: xRy iff $x \mid y$
$A = \mathbb{N}$: xRy iff $x \leq y$

Set A together with R is a directed graph $G = (V, E)$ with $V = A$, $E = R$.



Properties

A relation R on A is
* relfexive if $xRx \ \forall x \in A$
* symmetric if $xRy \Rightarrow yRx \ \forall x, y \in R$
* anti-symmetric if $xRy \underset{(\&)}{\wedge} yRx \Rightarrow x = y$.
* transitive if $xRy \wedge yRz \Rightarrow xRz$

67

Ex	Reflexive	Symmetric	Anti-Symm	Transitive	
$x \equiv y \pmod 5$	✓	✓	✗ (2,7)	✓	→ Equivalence
$x \mid y$	✓	✗ (1,2)	✓	✓	Partial Order
$x \leq y$	✓	✗ (1,2)	✓	✓	

⇒ An equivalence relation is reflexive, symmetric, transitive.

Ex: $A = \mathbb{Z}$: xRy iff $x = y$

$A = \mathbb{Z}$: xRy iff $x \equiv y \pmod n$ $n \in \mathbb{N}$

⇒ The equivalence class of $x \in A$ is the set of all elements in A related to x by R : denoted by $[x]$

$$[x] = \{y : xRy\}$$

Ex: $A = \mathbb{Z}$: xRy iff $x \equiv y \pmod 5$

$[7] = \{\ldots, -3, 2, 7, 12, 17, \ldots\}$

$[7] = [2] = [12] = [17] = \ldots$

68

A partition of A is a collection of disjoint nonempty sets $A_1, \ldots, A_n \subseteq A$ whose union is A.

Ex: $A = \mathbb{Z}$: xRy iff $x \equiv y \pmod 5$

$A_0 = \{\ldots -5, 0, 5, \ldots\}$

$A_1 = \{\ldots -4, 1, 6, \ldots\}$

$A_2 = \{\ldots -3, 2, 7, \ldots\}$

$A_3 = \{\ldots -2, 3, 8, \ldots\}$

$A_4 = \{\ldots -1, 4, 9, \ldots\}$

Theorem: The equivalence classes of equivalence relation on a set A form a partition of A.

Partial Orders

A relation is a (weak) partial order if it is reflexive, antisymmetric and transitive.

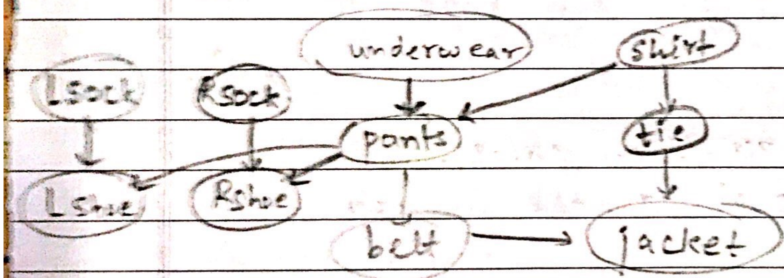
A relation is a (strong) partial order if it is irreflexive, antisymmetric and transitive (See book).

69

A partial order relation is denoted by $\preceq$ instead of R.

$(A, \leq)$ is actually called partial ordered set. or poset

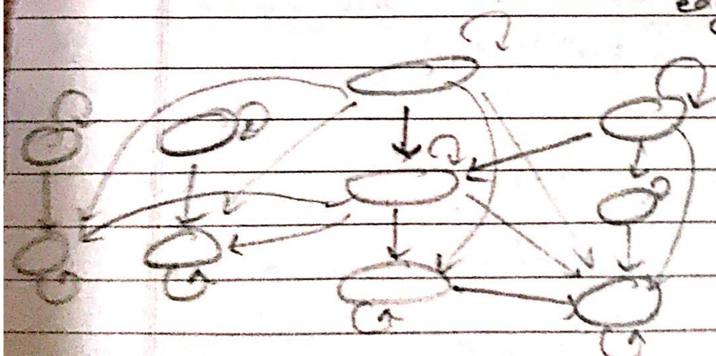
A poset is a directed graph with vertex set A and edge set $\preceq$.



(Hesse diagram)

Transitive $\Rightarrow$ No directed cycles

Antisymmetric $\Rightarrow$ Only one directed edges b/w two vertices



(Poset)

70

classmate
Date
Page

A Hasse diagram for a poset $(A, \preceq)$ is a directed graph with vertex set A and edge set $\preceq$ minus:
* all self-loops, and
* all edges implied by transitivity

<u>Theorem</u> A poset has no directed cycles other than self loops.

<u>Proof</u> By contradiction

Suppose $\exists$, $n (\geq 2)$ distinct elements $a_1, \ldots, a_n$ such that
$a_1 \preceq a_2 \preceq a_3 \ldots \preceq \underline{a_n \preceq a_1} \Rightarrow a_n \preceq a_1$ —②
(induction)

By transitivity, $a_1 \preceq a_2 \preceq \ldots \preceq a_n$
$\Rightarrow a_1 \preceq a_n$ —①

anti-symmetry

$a_1 = a_n$

Contradicts distinct element assumption. $\blacksquare$

71

classmate
Date
Page

Conclusion of the theorem

So, deleting self-loops from a poset makes a directed acyclic graph (DAG)

⇒ a and b are incomparable if neither $a \preceq b$ nor $b \preceq a$

⇒ a and b are comparable if $a \preceq b$ or $b \preceq a$.

⇒ A total order is a partial order in which every pair of elements is comparable.

Hesse diagram of a total order is a straight line.

A total order consistent with a partial order is called a Topological sort.

A topological of a poset $(A, \preceq)$ is a total order $(A, \preceq_T)$ such that

$$\preceq \subseteq \preceq_T$$

$$(x \preceq y \Rightarrow x \preceq_T y)$$

72

Theorem Every finite poset has a topological sort.

Proof $\Rightarrow$ $x \in A$ is minimal if there doesn't exist $y \in A$; $y \neq x$ such that $y \preceq x$.

$\Rightarrow$ $x \in A$ is maximal if $\neg \exists$ $y \in A, y \neq x$ s.t. $x \preceq y$.

not there exists

Lemma Every finite poset has a minimum element.

$\Rightarrow$ A chain is a sequence of distinct elements such that $a_1 \preceq a_2 \preceq \dots \preceq a_t$ (length of chain is t).

Proof Let $C = (a_1 \preceq a_2 \dots \preceq a_n)$ be the max length chain.

Case 1: $a \notin \{a_1, \dots, a_n\}$

if $a \preceq a_1$ then C is not the longest chain.

So, $\neg\, a \preceq a_1$

73

classmate
Date
Page 61

Case 2: $a \in \{a_1, \ldots, a_n\}$
if $a \preceq a_1$ then we have a directed cycle which contradicts Theorem ___.
So, $\neg\, a \preceq a_1$

$\therefore \forall\, a \in A \quad \neg\, a \preceq a_1$
So, by definition, a_1 is the minimum element.

Add Parallel Task Scheduling & Dilworth's Lemma.

L-13 (45:30)

Asymptotic Notation

tilde $f(x) \sim g(x)$ if $\lim_{x \to \infty} \frac{f(x)}{g(x)} = 0$

oh, big-oh $f(x) = O(g(x))$ if $\lim_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| < \infty$ (finite)

Multiple usage:

Formal math way

$f(x) \leq O(g(x))$; $f(x)$ is $O(g(x))$; $f(x) \in O(g(x))$

$O(g(x))$ is a set of all func. that grow slowly than $g(x)$

74

classmate
Date
Page 62

Theorem Let $f(x) = x$, $g(x) = x^2$
Then $f(x) \in O(g(x))$

Proof $\lim_{x \to \infty} \frac{x}{x^2} = 0$ which is finite

Theorem $x^2 \notin O(x)$

Proof $\lim_{x \to \infty} \frac{x^2}{x} = \lim_{x \to \infty} x$ is infinite.

Q. Is $x^2 \in O(10^6 x)$ true?
A. No.

Q. Is $10^6 x^2 \in O(x^2)$ true?
A. Yes

Q. Is $x^2 + 100x + 10^7 = O(x^2)$ true?
A. Yes

Theorem $x^{10} \in O(e^x)$

Proof $\lim_{x \to \infty} \frac{x^{10}}{e^x} = 0$ which is finite.
(L'Hopital's rule)

75

classmate
Date
Page 63

Theorem $4^x \notin O(2^x)$

Proof $\lim_{x \to \infty} \frac{4^x}{2^x} = \lim_{x \to \infty} 2^x \longrightarrow \infty$

Q. Is $10 \in O(1)$ true?
A. Yes

$f(x) \geq O(g(x))$ is meaningless.

omega $f(x) = \Omega(g(x))$ if $\lim_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| > 0$

Theorem $f(x) = O(g(x))$ iff $g(x) = \Omega(f(x))$

$f(x) \leq O(g(x))$ iff $g(x) \geq \Omega(f(x))$

$x^2 = \Omega(x)$
$2^x = \Omega(x^2)$
$\frac{x}{100} = \Omega(100x + 25)$

76

classmate
Date
Page 64

theta $f(x) = \Theta(g(x))$ if $0 < \lim_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| < \infty$

Theorem $f(x) = \Theta(g(x))$ iff $f(x) = O(g(x))$ & $f(x) = \Omega(g(x))$

$10x^3 - 20x + 1 \in \Theta(x^3)$

$\frac{x}{\ln(x)} \notin \Theta(x)$ | $\frac{x}{\ln(x)} \in O(x)$

$T(n) = \Theta(n^2)$ means T grows quadratically in n.

$O \Rightarrow \leq$
$\Omega \Rightarrow \geq$
$\Theta \Rightarrow =$
$o \Rightarrow <$ ($\leq$ not $=$)
$\omega \Rightarrow >$ ($\geq$ not $=$)

77

classmate
Date
Page 65

little oh $\quad f(x) = o(g(x)) \quad$ if $\lim_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| = 0$

little omega $\quad f(x) = \omega(g(x)) \quad$ iff $\lim_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| = \infty$

$$\frac{x}{\ln(x)} \in o(x)$$

$$\frac{x}{100} \notin o(x) \qquad \frac{x}{100} \in \Theta(x)$$

$$x^2 \in \omega(0)$$

<u>what not do</u>
<u>example.</u>