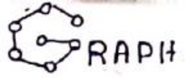
daughlus



TA : WED 2-4 LAB 202

Set:
A collection of distinct, <u>well-defined</u> objects. | computability vs truth
↳ membership status shouldn't be ambigious.

→ cartesian product: $n_{S_1} \times n_{S_2}$

→ Relation : any subset of cartesian product.

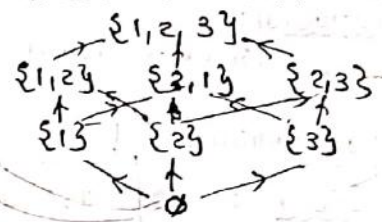
1) Reflective (IR) → $(a_i, a_i)\ \forall i$ must be in R.
2) SYMMETRIC (A)(ANTI): (ab) & (ba) ∈ R
3) Transitive : if (ab)(bc) ∈ R then (ac) ∈ R
4) Equivelence = Ref + SYM + TRA
5) Partial Order (Total order).
↳ Ref + AntiSYM + TRA

(D) (C)

no of relations = $2^{|D| \cdot |C|}$

IR Reflexive : none of $(a_i, a_i) \in R$.
Asymmetric : break atleast one sym.
Antisymmetric : cant take both. NAND

Partial order ≡ HASSE DIAGRAM

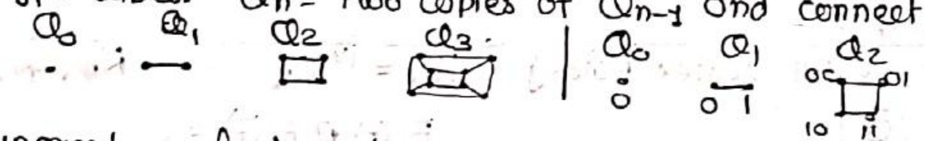


→ Equivelence is closely related to Partition.
→ Every partial order is a DAG, but
Every DAG need not to be partial order.
→ Hasse diagram is minimal representation
and partial order is maximal representation (includes transitive edges).
→ A directed graph is acyclic iff its transitive closure is antisymmetric.

* Hypercubes are hasse diagram of power set partial order
→ any directed acyclic graph is subgraph of power set partial order
Algo : while (whole graph is not labled) {
find node whose all in nodes are labled
lable it
}
{label sources, recurse, introduce if same)

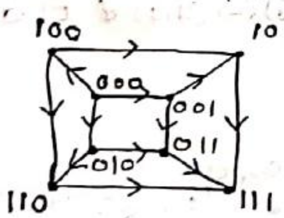
Q Embed DAG in power set partial order (smallest). [Graph Embedding.]

* Hypercubes: Q_n = two copies of Q_{n-1} and connect corrosponding edges



Pre concomitoe with copy numbes {0,1}.

→ Hypercube of dimention n is the hasse diagram of power set partial order of set of n elements.



→ draw edges to 1 from 0 in same position.

→ strongly connected components in directed graph is an equivelence relation. and strongly connected component is called kernel.
→ Rechability in undirected graph is an equivelence relation.
→ Strongly connected graph ⇒ only one strongly connected component.
→ for DAG, we show only one vertex per equivelence class. (kernel)
→ Graph genrated by putting one vertex for equivelence class is called kernel.

Automorphism

- functions are special class of Relations.

$f: N \to N$, $f(x) = x^2$ | Range is subset of co-domain, All points which has pre-image.

D, C

Total Relations $= 2^{|D| \times |C|}$

- Injective (one-one) function :- $|D| \leq |C|$

$\forall\ x, y \in D,\quad x \neq y \Rightarrow f(x) \neq f(y)$

Bijective = one one + onto

- Surjective (onto) function :- if $R = C$. $|D| \geq |C|$

$|D| = |C|$

- Bijective functions on a set is called Permutation.

* function composition:

$g(f(x)) \qquad R_f \subseteq D_g$

* Order of a group:

How many times apply operation to come back.

$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 4 & 5 & 6 & 7 & 3 \end{pmatrix}$ — 2, 5

$\text{lcm}(2, 5) = 10$.

* falling factorial:

n, m ... $m \times m-1 \times \cdots m-n+1$

for Bijective fuction : $n!$

* Simple Graph : forbids self loops and multiple edges.

"A finite graph is a combination of a finite set V of vertices and a binary, irreflexive, symmetric relation on V, called E"

→ complete graphs are transitive too. └ Doesnt relate any element to itself.

* subgraph: $G(V, E)$

H is a subgraph of G if H is a graph and $V' \subseteq V$, $E' \subseteq E$

* spanning subgraph: $V' = V$

→ no of spanning subgraphs $= 2^{|E|}$

* induced subgraph: edge maximal subgraph for any specified vertex set.

→ only one subgraph on any vertex set.

→ no of induced subgraphs $= 2^{|V|}$

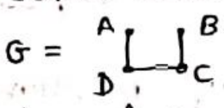
* Complete graph :- $E = \{\text{All vertex Pairs}\}$, $|E| = {}^{|V|}C_2$ $\quad K_n, C_N, P_N$

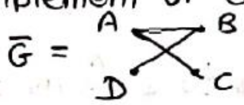
* Graph Complement :- $(\bar{G})$

same vertices, complement of edge set.

$E(G) \cap E(\bar{G}) = \emptyset$

$E(G) \cup E(\bar{G}) = K_V$

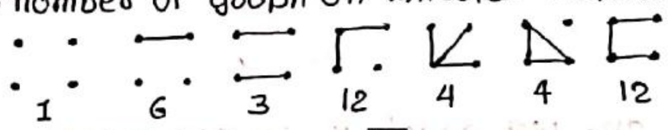




$(u, v) \in E(G) \Leftrightarrow (u, v) \notin E(\bar{G})$

⇒ number of graph on n labled vertices $= 2^{{}^nC_2}$

⇒ number of graph on unlabled vertices : Bruteforce.



no of grap with >4 edges = no of graph with $5(6-4) \leq 2$ edges

* Edge less graph : $\overline{K_n}$

* Path: P_n

* Cycle : C_n

⇒ If two graphs are isomorphic, thier complements are isomorphic. only if complement is taken with respect to complete graph.

* Self complementary graph: A graph which is isomorphic to its complement

$C_5 \cong \overline{C_5}$, $P_4 \cong \overline{P_4}$

* structural properties : only structure independent of label.

↳ Infinite. All structural properties are identical.

[Quazi Polynomial -]

* ISOMORPHIC:

"G is isomorphic to H iff there exists a bijective function $f: V(G) \to V(H)$ such that $(u,v) \in E(G) \Leftrightarrow (f(u), f(v)) \in E(H)$."

→ function f is called isomorphism.

Thm: for two graphs G, H, $G \cong H \Leftrightarrow \overline{G} \cong \overline{H}$

* AUTOMORPHISM:

"an automorphism is a bijective function f from a vertex set of a graph G to itself. such that $(u,v) \in E(G) \Leftrightarrow (f(u), f(v)) \in E(G)$.

→ isomorphism and vertex sets of automorphism are equivelence relation.

Reflexive → identity

Symmetric → inverse

transitive → composition

$$G_1 \xrightarrow{f} G_2 \xrightarrow{g} G_3$$

$$(u,v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2 \Leftrightarrow (g(f(u)), g(f(v))) \in E_3$$

* Regular Graph: every vertex has the same degree.

2-Regular graphs ⇒ disjoint cycles.

→ set of all automorphisms forms a 'Group.'

"Binary relation between vertices based on existance of an automorphism between any pair is an equivelence relation."

→ automorphisms are same for complement graphs.

→ no of automorphism on complete graphs are **n!**

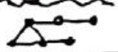
→ no of automorphisms for path = 2 , no of eq. class $\lceil n/2 \rceil$

cycle = 2n, 1.

* Vertex transitive graph : all vertices are same. : 1 equivelenc class size n

* Rigid graph : all vertices are different : n equivelence class size n

only one automorphism.



* vertex transitive ⇒ reguler.

* Knesser Graph: KN(n, k) , n ≥ k

Disjointness Graph

complement

Intersection Graphs

Knesser graph ⊂ Disjointness graph

Interval graph ⊂ Intersection graph.

KN(n,k) : n = set cardinality

k = subset sizes

vertices = $\binom{n}{k}$

edges = $\dfrac{\binom{n}{k}\binom{n-k}{k}}{2}$

"one vertex for each subset of size k of a set of size n"

Adjencecy ≡ Disjointness

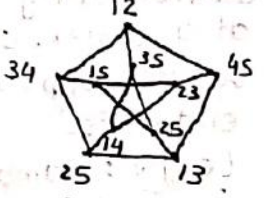
Matching : k = n/2

degree of any vertex : $\binom{n-k}{k}$

If n = 2k + 1, its called "ODD GRAPH".

Peterson graph is an odd graph with k = 2.

Peterson graph is edge transitive on P_4.



Peterson graph.

↳ vertex transitive

↳ edge transitive

↳ P_4 transitive

* Edge transitive: all edges are same.

"There exists an automorphism where we can map vertices of any edge to any other edge."

⇐ Vertex transitive but not edge transitive.

⇐ not vertex transitive but edge transitive.

no of automorphisms ⇒ n! × m!

$K_{2,3}$

$K_{m,n}$

* Cutvertex: an edge is cutedge if it does not belong to cycle.
* maximal path within a graph: cant extend endpoints of existing path.
  → endpoint of maximal path cant be cutvertex.
  → either all or none vertex are cutvertex in ~~edge~~ vertex transitive graph.
  → every graph has atleast two non-cutvertexes.

Triangle transitive
K3 transitive

DECOMPOSITIONS

→ Identical, similar, arbitrary
ex. self-complementary, m cycles
def: partition on edge set.

$$d_G(v) = \sum_{i=1}^{k} d_{G_i}(v)$$

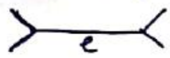
→ partition K_6 into triangles. original degree = 5, new degree = multiple of 2 thus not possible.

→ self complementary: decomposition of K in two identical graphs.
  no of ~~ver~~ edges in $K_n = \frac{n(n-1)}{2}$.

⇐ generalize this.

Petersson graph doesnt have 4 length cycle, but any 2 vertices has unique 2 path.

Q How many 5 cycle in petersson graph?



for this edge, 4 possibilities for 3 length path.
for two end vertices has unique completion.
thus each edge has 4, 5-cycles. total 15 edges.
thus total 60 cycles. but one cycle is counted in all 5 edges
thus no of 5 cycles = 12.

Petersson graph has induced 6-cycle when we remove 6 cycle and corresponding edges remaining is $\Lambda(K_{1,3})$ there are total 10 such graphs. thus total no of 6-cycles are 10.

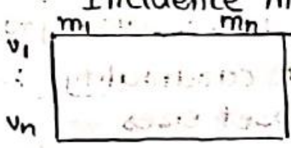
ADJACENCY MATRIX:

first theorm of graph theory:

$$\boxed{\sum_{v \in V} d(v) = 2|E|}$$

Handshaking Lemma.

Incidence Matrix:-



walk: "an alternating sequence of vertices and edges, begins and ends at a vertex. such that for each edge in the sequence, the vertices appearing immediately before and after it are its two endpoints."

Length: number of occurences of edges in sequence including dups.
odd walk and even walk.
closed walk: first vertex is same as last vertex.
open walk: not-a-closed walk.

Trail: "A trail is a walk without repeating edges"

Path: "A path is a trail without repeating vertices" | Induced Path

Cycle: "A cycle is a walk without repeating vertices except start vertex."

A shortest path is induced, but induced path may not be shortest.

Scanned by CamScanner

→ every closed odd walk contains odd cycle.

$A^2[i][j]$ = common neighbours of i and j.

Principal diagonal gives degree sequence.

– no of walks of length of 2.

→ t^{th} power of adjecency matrix's entry i,j represents the number of walks of length t between i and j.

→ minimum degree k ⇒ length of path k, cycle of length atleast $k+1$

Decompositions :- Eulerian Trails : closed trails which cover all edges.

: closed spanning trail

$d_G(v) = d_T(v)$

– degree of interior vertex of trail ⇒ even

closed trail endpoints degree ⇒ even

open trail endpoints degree ⇒ odd

1 → If graph contains an vertex with odd degree, it cannot have eularian cycle. because to get a closed trail, all vertices must hve even degree.

2 → graph must be connected, only one non trivial component.

↳ necessary condition for a graph to have an eularian cycle.

1) every vertex has even degree

2) only one non trivial component.

Algo: start with any vertex, find a tour and come back to same vertex if some edges are remaining find another tour. concatenate this path in first string.

Q decompose K_7 in cycles. for K_{prime} → start with 1 +1 ... mod prime.

2 ... mod prime ...

Reconstruct a graph, complements, no of graphs with certain property.

* Bipartite graph : R-partite Graph.

NP complete

– Independent set : vertices with no interconnect edges. } complements

CLIQUE : induced complete subgraph

↳ induced subgraphs with zero edges.

Largest of Independent set size $= \alpha$

Maximum clique $= \omega$

$\alpha(G) = \omega(\bar{G})$

$\alpha \cap \omega = 1$

$\alpha + \omega \leq n+1$

* single vertex is independent set and clique.

– edge less graph is only graph which is one-partite.

Graph	op	status
BIP	super	may not remain
not	super	may not
BIP	sub	remains
not	sub	may become

→ Bipartite-ness is closed under subgraph operation.

→ A graph is bipartite iff it does not contain an odd length cycle.

Horizontal cross edge in BFS denotes odd cycle.

→ If there is an odd cycle then there is an induced odd cycles.

* Algo to find odd cycle :

1) Run BFS

2) for each $(x,y) = e \in E(G)$:

3) if $\pi(x) \neq y$ & $\pi(y) \neq x$;

4) if $d(x) = d(y)$:

5) return non-bipartite

6) Return Bipartite

→ If graph is not bipartite, finding subgraph maximum which is bipartite

Approximation Algorithm to find maximal bipartite subgraph.

→ start with random bipartition

→ for each vertex :

if it has more neighbour in own part than other

move it to other part.

$$d_G(v) = d_e(v) + d_f(v)$$

$$d_f(v) \geq \frac{d_G(v)}{2} \Rightarrow \Sigma d_f(G) \geq \frac{1}{2} \Sigma d_G(v)$$

$$\therefore 2|E_B| \geq |E_G|$$

$$\therefore \boxed{|E_B| \geq \frac{1}{2}|E_G|}$$

Algo 2: start with both empty parts : bring new vertices : put in with least neighbours.

→ Every Bipartite graph has unique bipartition. (connected)

---

* vertex degrees: no of edges endpoints touching a vertex.

degree sequence : $d_1, d_2, \ldots d_n$

minimum degree of graph = δ

maximum degree of graph = Δ

graph reconstruction:

Q. given n subgraphs of a graph by deleting each vertex. Find original graph.

[degeneracy / Tree width]

{ Δ } { •—• } { •—• } { •—• • }

level 1 : Construct a graph from degree sequence with multiple edges & selfloops.

- only requirement is that Σd_i is even. we can always construct a graph.

1) for all odd degree vertices draw edge choosing two.

2) selfloops.

level 2: Construct simple graph.

requirements : sum must be even, max degree can be $(n-1)$.

0 and $(n-1)$ cannot be appear at same time in sequence

thus atmost $(n-1)$ different values possible.

Degeneracy : minimum over all ordering [max left degree]

k-degenerate : every round delete vertex with degree at most k and graph can be destroyed.

Tree is 1-degenerate. | Planar graphs are 5-degenerate.

To make ordering:

→ choose a vertex with left degree $\leq k$.

→ place it at rightmost position, ignore it, repeat.

If we dont know degeneracy use minimum degree every time

→ For regular graph, degeneracy = degree.

[Maximum value of minimum degree over all subgraphs]

left

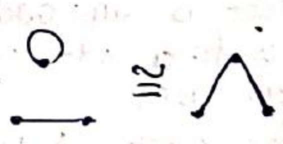
* A & B are sets

$|A| > |B|$, $A \supset B \Rightarrow \exists a$ s t $a \in A$ & $a \notin B$.

If sets are vertex neighbourhoods then its degrees.

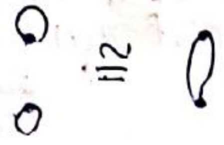
level 3: multigraph, allow multiple edges no loops.

we can only eliminate loop iff there are two disjoint loop or a loop and disjoint edge.



Algo : construct level 1 graph.

eliminate loops.



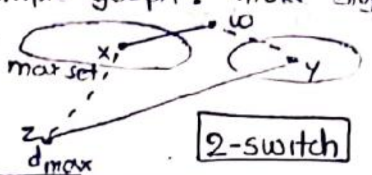
if $d_1 d_2 \ldots d_n$ is nonincreasing order then

$(\Sigma d_i)/2 = \partial$ & $d_1 \leq \sum_{2}^{n} d_i$ then only graph is possible.

Ex: 13, 9, 8, 6, 6, 4, 4   labels denotes number of loops
remove some number of loops

for simple graph: make biggest degree vertex adjacent to other highest degree vertex.

Proof:



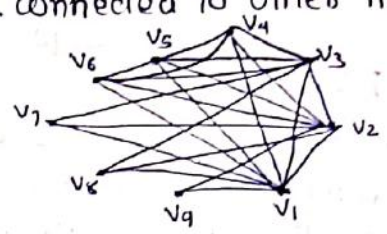
if biggest degree vertex z adjacent to y and not x then, $d_x > d_y$ they there exists w such that (xw) exists and (wy) doesn't exist.
then replace (xw)(zy) with (xz)(wy). degree remains same.

Havel-Hakimi Theorem:-

- write degrees in non increasing
- make highest degree vertex connected to other highest degree vertices.

Ex:

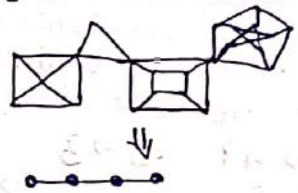
v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
8	8	7	5	5	5	3	3	2
	7	6	4	4	4	2	2	1
		5	3	3	3	1	1	0
			2	2	2	0	0	0
				1	1	0	0	0
					0	0	0	0



Directed Graph:-

→ orientation: giving direction. For m edges there are 2^m orientations.
Total indegree = total outdegree.
Orientation of undirected graph is XOR for every edge. either of two direction present.
Orientations of Complete Graphs are called "Tournaments".

* Block: maximal subgraph with zero cut vertices. for undirected.
  Block cut point tree:
  Graph constructed by undirected edges
  replace block with ~~edges~~ vertices and cut vertex with edges



Disconnected graph is not k-connected for any k.
Block is 2 connected subgraph.
If graph is k-connected then it is also 1, 2, ..., k-1 connected.

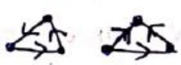
k-connected graph: it is impossible to make graph disconnected by removing k-1 vertices.

cutvertex: after removal of vertex, number of connected components increases by 1 or more.

strongly connected component: all ~~apt~~ ~~co~~ pair of vertices are two way reachable.

→ If an undirected graph has cut edge then there is no possible way to give direction to edges such that resulting directed graph is strongly connected.

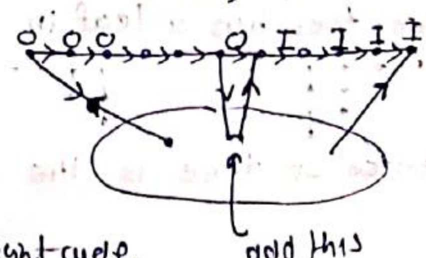
→ every tournament has a hamiltonian path.

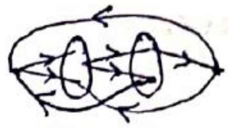


→ If there is a cycle, then there is a triangle.

→ There exists a vertex from which every other vertex can be reached in atmost two steps. (KING).

if dir of e is ↑ then found C_3 else use e in left cycle and forget right cycle.

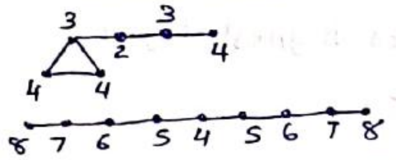




→ Other then complete graph, are there any other graph, for which any orientation has a king?
→ what are the graph familier for which no matter how you orient them there always exists a vertex from which any other vertex is at atmost distance k.

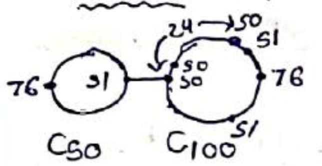
* **Distance** :- The length of the shortest path between u & v.
(number of edges.
→ If there are no path then distance is ∞.

* **Eccentricity** :- $ecc(u) = \max_{v \in V} d(u,v)$



* eccentricity sequence : sequence of eccentricites of all vertices.

* **Radius** : minimum value of ecc of a graph.
* **diameter** : maximum value of ecc of a graph.
} Triple optimization.
* **center** : set of vertices with minimum ecc.



Radius : 50
diameter : 76

If graph has vertex with ecc z, then graph cannot have vertex with ecc < $\frac{z}{2}$ or > 2z.

$\forall (u,v) \in G : |\varepsilon(u) - \varepsilon(v)| \le 1$

If graph has radius r, then for any two vertex u and v thier distance from center can be maximum r thus. thier distance cannot be more then 2r.

→ There are atleast 2 vertices with ecc more then minimum.

$e_{50} \Rightarrow 49$ $\quad e_{51} \Rightarrow 3$
$e_{76} \Rightarrow 2$ $\quad e_{52 \ldots 75} \Rightarrow 4$

$24 \times 4 + 3 + 2 + 49 = 150.$
96

→ If we add more edges, term by term ecc. either remains same or decr

* **Trees** :- Connected acyclic undirected graph.
→ no of new cycles created when adding an edge between x and y is same as no of paths between x and y.
→ adding an edge can either add cycle xor reduce no of componen
→ adding an edge can reduce at most one component.
→ If any pair has more then one path then there exists a cycle. thus in tree, every pair of vertices has unique path.

Thm : An edge is a cut-edge (unifying edge) iff it does not lie on a cycle.
If an edge lies on cycle, we remove it and add it back then it adds a cycle. thus it cannot incresed number of component.

→ In tree, every edge is a cut edge.
→ no. of maximal paths = no of leaves C_2
→ A tree with degree k, it has k leaves.
→ every tree has a leaf in larger partites set in bipartition.

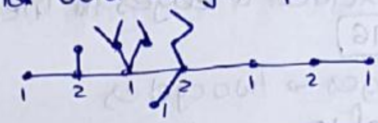
$d(v) + e(v) = n-1$

– longest path, Center can be either on vertex or one edge (two vertices).

* Center of tree is the center of the longest path in tree.

→ larger partite set always has the leaf.
→ If there is and odd length path then both partite set has leaf.
even path.

maximal path



imbalance is either remains same or increses. as we introduce vertices.

× Caterpiller Graphs : one path and all other nodes are at distance 1 from path.

→ in path, leaf-partite set has higher number of leaves.

→ n rooted tree if some level has no leaves then next level has same or equal numer of nodes

→ all leaves are at $\boxed{\text{odd level}}$ or at even level.

$|l_1| \geq |l_0|$, $|l_2| \geq |l_1'|$ (non leaf nodes), $|l_3| \geq |l_2|$

If leaves are at odd level, then we want to proov,
$|l_1| + |l_3| + \cdots + |l_{2k+1}| > |l_0| + |l_2| + \cdots + |l_{2k}|$

saparate odd levels in leaf nodes & non leaf nodes.
$|l_1'| + |l_1''| + |l_3'| + |l_3''| + \cdots + |l_{2k+1}'| + |l_{2k+1}''| > |l_0| + |l_2| + \cdots + |l_{2k}|$

there are no leaves at leven. thus $|l_2| = |l_1'|$, $|l_{2i+1}| \geq |l_{2i}|$
there are no ~~leave~~ leaf nodes at odd level = no of leaves at previous odd level.
non leaf nodes at odd ~~long~~ level are greater.
thus no of nodes at odd level are greater.
thus larger partite set always has leaf.

Q Any tree on k edges can be embeded in graph with mimimum degree k.
→ take any vertex of tree → map to any vertex on graph.
take a new node adjecent to mapped node, there is always vertex with which we can map it to.

Numbeo of trees on N vertices. | no of graphs = $2^{{}^nC_2}$
→ counting labled graphs is easier then counting unlabled graphs. | labled no of trees = n^{n-2}

→ Prufer code : bijective function with tree.

do {look at current tree
find lowest indexed tree, remove that
write down its label neighbors } (while 2 vertices left)

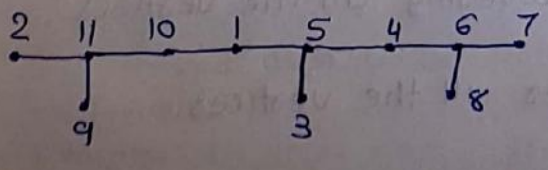
[11, 5, 6, 6, 4, 5, 1, 10, 11] ← Prufer code (len = n-2)

I[1, 10, 11, 5, 4, 6]
L[2, 9, 3, 7, 8]

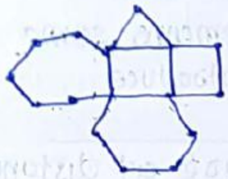
All leaves in original tree does not appear in code.
→ number of times a node appears = its degree − 1
if all elements are unique in code → Path
if all elements are same in code → star

code → tree : vertices = len + 2

→ find smallest missing number except set, so leaf adjecent to first element in code
Enter leaf in set.



* Counting spanning trees of graph:-



→ 16 cycle outside, exclued 4 edges in inner cycle. $\Rightarrow$ 16 cycle gives = $\boxed{16}$ ... 16

→ take one of inner edges → two cycles

any of 6 × any of 10 = 6×10 = 60

2 × 14 = 28

3 × 13 = 39

5 × 11 = 55

Total: 198

→ take two inner

1) adjecent ones;

2×6×8 = 96

2×3×11 = 66

3×5×8 = 120

5×6×5 = 150

Total: 432

2) non-opposit

7×3×6 = 126

2×5×9 = 90

Total: 216

→ 3 edges included

4× [2×3×5×6] = 4×180 = 720

sum = 198 + 432 + 216 + 720

$\boxed{\text{sum} = 1566}$ ≈ 700 $\boxed{\text{WINNER}}$

* MATCHING: size of matching = no of edges | semi reguler Bipartile graph

Collection of pairwise non-adjecent edges.

Maximal matching : set of edges such that no superset is matching. (local optimal).

↳ without deletion, no augmentation.

The size of a maximum matching = $\boxed{\alpha'}$

maximum independent set = α

maximum clique = ω

Induced matching : distance between any two selected edges is atleast 2.

Perfect matching : all vertices are covered in matching. [all saturated]

Saturated and unsaturated vertices for matching on graph.

Number of perfect matching on K_{2n} = $\frac{2n!}{2^n\, n!}$ = $(2n-1)(2n-3)\cdots(1)$

° Berge's theorm : a matching is maximum if it des not have augmenting Path.

Augmenting path : path on matching, starts and ends at unsaturated vertices and flip all edges.

* VERTEX COVER & EDGE COLOR :-

↑ set of vertices such that atleast any one edpoint of edge is present.

→ These are cross concepts, blocking other while covering one.

NPH ⇒ Vertex cover : "set of vertices covering all the edges"

easy edge cover : "set of edges covering all the vertices".

NPH ⇒ Dominating set : (NP-Hard)

subset of vertices that cover all the vertices.

Independent set :

subset of vertices with no two adjecent.

Matching : subset of edges with no two adjecent.

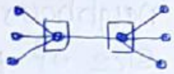
← dominating set, but not a veatex cover.

→ every graph does not have edge cover. (graphs with degree 0 veatex).

min veatex cover : β
min edge cover : β'
dominating set : γ

largest independent set $= \alpha$
maximum matching $= \alpha'$

← minimal dominating set



← minimum dominating set

→ lower dominating set = lowest cardinality of minimal dominating set.

← forms an independent set.
← cannot have edge here.

for a bipartite graphs, $\beta = \alpha'$.

$\alpha + \beta = n$
$\beta \geq \alpha'$ | Konig's thm
$\alpha' + \beta' = n$
$\beta \leq 2\alpha'$

→ greedy algorithm for maximal matching is factor 2 algorithm.

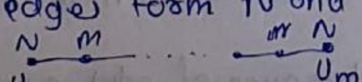
→ for any maximal matching, we can return all veatices as veatex cover.
Thus $\beta \leq 2\times$ maximal matching $(2\alpha'')$
$\therefore \boxed{\beta \leq 2\alpha'}$

| Proof by contradiction.

→ If we assume that maximal matching is less then half of α' then,
$2\alpha'' < \alpha'$
$\therefore \beta \leq 2\alpha'' < \alpha'$
$\therefore \beta < \alpha'$, this is contradiction to Konig's theorm.

→ If there is an augmenting path, then matching is not maximum.
absence of such path means matching is maximum.
no augmenting path $\longleftrightarrow$ maximum matching.

Berge's theorm:
Proof : take matching M with no ~~aug~~ augmenting path.
suppose there exists some matching N, such that $|N| > |M|$
take XOR of edges in both sets. this gives degrees 0, 1 and 2. they are either isolated, paths or cycles.
we have thrown away equal numbers $\therefore$ XOR $= N \cup M - N \cap M$
thus still $|N| > |M|$. two matchings cannot cover odd cycle.
thus cycles are even length. which has same number of edges from N and M. still $|N| > |M|$. there are pths remaining.



found augmenting path ~~N~~ for M. this contradicts assumption.

Algo: for bipartite:
1) find maximal by crude way.
2) find augmenting path using BFS from unsaturated. flip all the edges.

Perfect matching in bipartite graph:-

$|L| = |R|$ if $|L| \neq |R|$ then we can find matching in subgraph.

Hall's theorem & Hall's Condition:-

degree of $u = 1$

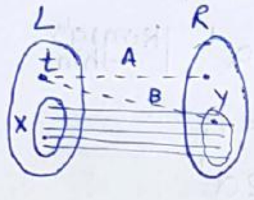
neighbourhood of $\{u, v\} \in [2, R]$

"for any subset of L neighbourhood of size of neighbourhood is always greater than or equal to size of L"

Halls Condition $\Rightarrow \forall X \subseteq |L|, |N(X)| \geq |X|$ L is smaller size set

Proof: Assume Halls condition hold on left.

there is a largest matching in L. if L is not saturated



then we can find augmenting path. starting from unsaturated vertex from L to R.

since halls condition holds, and $|X| = |Y|$

means $|L| \leq |R|$.

now take $X \cup \{t\}$,

A] 1) if t has a neighbour in R, then add it to matching.

2) start BFS from t, find augmenting path.

which contradicts assumption that this is biggest set.

B] if neighbour of t is in Y,

{in first level find unmatched edge,

second level find matched edge } loop for all even-odd.

eventually we will reach a node outside of Y. thus we have found and augmenting path.

Colorary

$\Rightarrow$ every regular bipartite graph has a perfect matching. $\Rightarrow |L| = |R|$

Proof by contradiction: $\exists X \subseteq L, |N(X)| < |X|$

number of edges in $N(X) = r|X|$. these edges goes to set of size $|N(X)|$

average degree going out $= \frac{r|X|}{N(X)} \Rightarrow |X| = N(X)$

Contradiction.

$\rightarrow$ every bipartite graph is Δ-edge colourable.

Thm: Bipartite graph of supergraph of bipartite graph has matching.

make copy B' of B.

connect vertices of ~~sub~~ maximum degree vertices. max remains same minimum degree goes up.

we have to show new is bipartite $\rightarrow$ absence of odd cycles.

By hypothesis B & B' doesn't have odd cycle.

To find odd cycle it must be crossovers, but number of crossover edges are even. and both sets are exactly identical. thus there are no odd cycles. hence.

"Every bipartite graph has Δ regular bipartite supergraph, which has perfect matching."

we can eliminate entire graph with Δ matchings. because if we remove a matching from graph (r regular) it becomes r-1 regular.

* Perfect Matching:-

→ number of vertices must be even in each component.

→ minimum degree > 0.

→ If some vertex has degree n-1, it will always be mapped.

These vertices are called universal vertices.

Tutte's Theorem:

A graph has a perfect matching iff Tutte's condition holds.

Tutte's Condition:

$\forall S \subseteq V$, the number of components with odd vertices in $G[V \setminus S]$ is $\leq |S|$.

→ addition and removal of vertex can make graph connected.

→ If a graph holds tuttes condition, its supergraph created by adding edges also holds tuttes condition.

Proof: there is some set for which number of odd component is more. the new edge added can be in any component. one edge addition can decrease no of component by 1. if new edge is in odd component then we must have merges (odd + even). thus number of odd component must remain same. If new edge is in even component then it must have been formed from (even + even) thus # supergraph also holds.

Thus, tuttes condition holds on edge addition operation.

Scanned by CamScanner