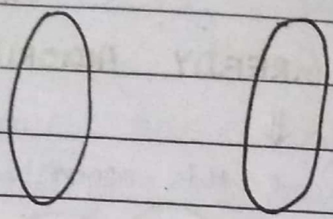
Bipartite Graphs.

Generalisation R-Partite Graphs.



Partition of vertex set
at most two independent sets.

- Independent Set: Subset ~~Decomposition~~ of vertex set
  st induced subgraphs. has no edges
  $\Rightarrow \alpha$.

- Clique: Set of vertices such that all edges have
  edge between them.

→ If a graph has a clique & independent set, their
intersection can have at most 1 vertex.

Sum of the size of $\underbrace{\text{largest}}_{\alpha}$ independent set &
$\underbrace{\text{largest clique}}_{\beta}$ size

Maximal ⇒ There is no super-set better solⁿ of
current state. There can be bigger set which
has a better solⁿ but for that, we have to go
back a step & continue, so not a super set.

$\boxed{\alpha + \beta \leq n+1}$ Subset of set can't be more than
total set. Intersect is 1.

Complete graph of n vertices

$$\boxed{\begin{matrix}\alpha = 1 \\ \beta = n\end{matrix}}$$

→ A graph is bipartite if we can partition the vertex set in at most 2 independent sets.

→ Every connected bipartite graph has unique partitioning

* Closure Property

super graph → adding edge / vertex
sub graph

Given a graph can or can't be bipartite

Bipartite
- Super: Anything
- Sub: Bip.

Non-bipartite
- Super: Non.
- Sub: Anything

→ Bipartite is closed under sub-graph operation
Non-Bipartite is closed under super-graph operation

⇒ Graph is bipartite if & only if it does not contain odd length cycle.

To show a graph is not-bipartite, we can just show one subgraph that is not-bipartite
also can show an odd cycle.

⇒ Algo to partition:
Start from one vertex v_1, adj of $v_1 \to v_2$ goes to another partition, its adj $\to v_3$ to first & so on.

If there exists a horizontal edge ⇒ Odd cycle
vertical edge ⇒ even cycle

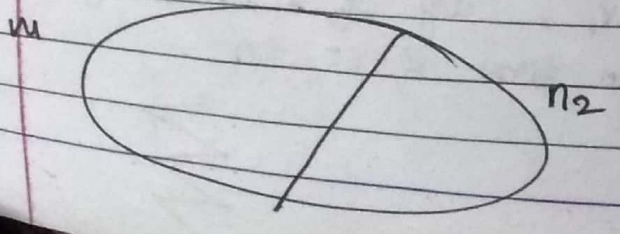
First Run BFS
Find parent, dist
~~for~~ every edge if (u,v) if par(u) = v or par(v) = u,
then (u,v) → tree edge.
else cross edge
↳ if dist(u) = dist(v)
↳ horizontal cross edge
Non bipartite.
else |dist(u) − dist(v)| = 1
↳ vertical cross edge
continue.

If not horizontal, all edge is tree or vertical.
~~ver~~ +1 or −1 movement
ie if we gofrom u → v
so can't reach same vertex in odd steps.

If there is an odd cycle then it must be induced odd cycle

⇒ Shortest cycle is induced.

If there is odd cycle & its not induce that means a chord exist

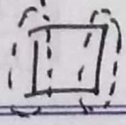


~~One~~ of Only possibilities
n_1 → odd & n_2 → even
n_1 → even & n_2 → odd

even n + 1 → odd
so decomposition in shorteel odd & even cycle.

Scanned by CamScanner

Even if graph is bipartite, not all partitions are valid.

eg: 

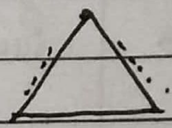
1) Run BFS (parent, dist)
2) if par(u) = v or par(v) = u → tree edge (for all edge)

else cross edge.

↳ if dist(u) = dist(v) → odd cycle

then horizontal cross edge

Non - bipartite



else |dist(u) - dist(v)| == 1

vertical cross edge

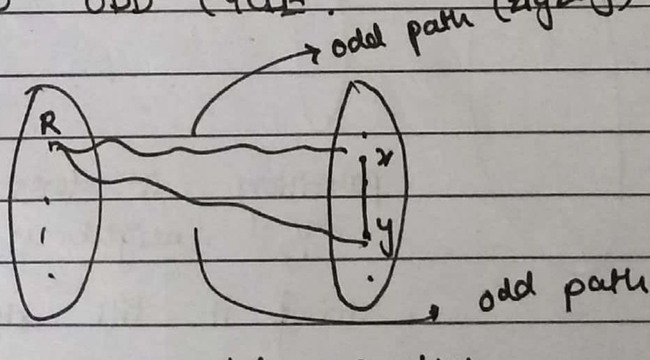
continue.

Proof: Graph is bipartite when no odd cycle.

Take two independent empty sets. Take an arbitrary vertex, apply BFS.

Put level i in one set & i+1 in other

Assume: NO ODD CYCLE.



~~Assume~~ If not valid partition

then we'll have closed odd path (odd cycle)

But we assumed no odd cycle.

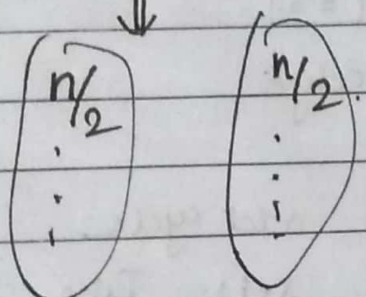
So contradiction.

Hence proved.

Finding a subgraph which is Bipartite (easy) but finding largest subgraph (NP complete).

↳ Approximation Algorithm. (2 factor Algo)
Upper Bound : Graph itself.

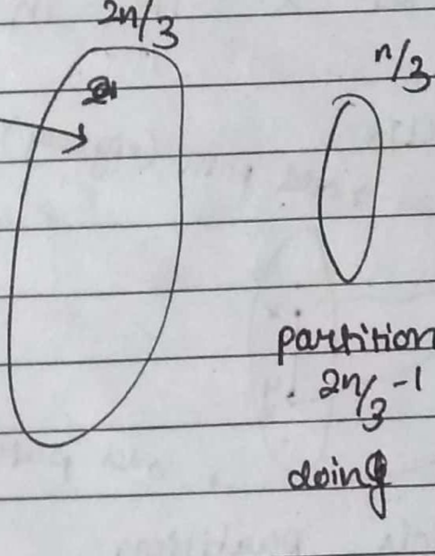
⇒ Partition of Complete graph
⇓



each vertex degree = $n/2$.
So total edges = $n^2/4$.
50%.

If we partition in $n/3$ & $2n/3$

Each vertex here has degree $n/3$
so if we move vertex to other



partition, it looses $n/3$ ~~edges~~ neighbour & gain $2n/3 - 1$ neighbour. Beneficial. Keep doing it till stable ($n/2$ & $n/2$.

$$d_G(u) = d_{local}(u) + d_{foreign}(u)$$
↳ in original graph

If bipartite
$$d_I(u) \geq \frac{d_{Graph}(u)}{2}.$$
↳ If not, move on other set so $d_I(u) \geq \frac{d_G(u)}{2}$.

in bipartite subgraph

So total degree_n is $\frac{1}{2}$ than original degree

Hence. edges in bipartite is at least $\frac{1}{2}$ than original graph

$$d_I(u) \geq \frac{d_G(u)}{2}.$$

$$\sum d_I(u) \geq \sum \frac{d_G(u)}{2}$$

$$\Downarrow$$

$$\Rightarrow \quad 2|E_B| \geq \frac{2|E|}{2}.$$

$$\therefore |E_B| \geq \frac{|E|}{2}.$$

$|E_B| \leq |E|$

↳ can't be more than original

So algo, take an order. one by one add it to the set which has least neighbour of the vertex

Take an order of vertex. $d_l \rightarrow$ left degree $d_r \rightarrow$ right.

$$d_G(u) = d_l(u) + d_r(u)$$

At each point, when entering, the vertex gains at least $\frac{d_l(u)}{2}$ (neighbours)

So at every add^n, total degree addition will be $2 \times \frac{d_l}{2} \rightarrow d_l$.

Every connected bipartite graph has unique bipartitions

Vertex Degrees Section 1.3

⇓

no. of neighbours

if multigraph → include multiplicity

self loop contributes −2

Degree Sequence

$d_1, d_2, d_3 \ldots d_n$

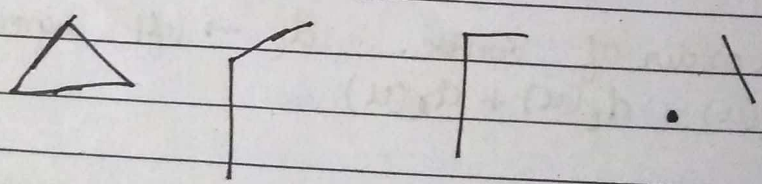
In-degree, Out-degree → Directed graph

Degeneracy tree-width.

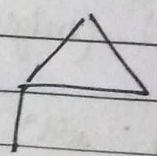
⇒ There exists degree sequences for which no graph is possible.

Reconstruction Problems.

⇒ Given: ~~there~~ are 4 vertices & 4 ~~each having 3~~ subgraphs are given. Can you reconstruct the original graph



Original graph



⇒ Given a potential vertex seq, can you construct the graph?

Case 1: graph with self loops & parallel edges → easy

Original degree − used up degree = Residual degree

Graph ke notes ki!?

Page No.
Date :

⇒ Only req is total is even (allowing self loops & parallel edges)

⇓

make arbitrary pairs of all the odd degree vertex (even number of odd degree vertices)

Then self loops.

What is the conditions for degree sequence of simple graph?

any vertex can have 0 to n−1 degree.

⇒ But 0 & n−1 degree vertex can't be both present

Because 0 means no neighbours, & n−1 means all neighbours.

So n values possible but 0 & n−1 together not possible

⇒ There has to be at least two vertices with same degree (Pigeon hole principle — n different values for n degrees but 0 & n−1 can't appear at same time so at least two have same degrees)

⇒ When you are removing a vertex degree, you are not just removing the vertex, you are reducing the degrees of neighbour with 1.

Q If you remove the maximum degree vertex, avg drops

Q If you remove the minimum degree vertex, avg may drop ~~of~~ ~~...~~ ~~...~~. { Here $\frac{avg}{n-1}$ → almost same

Degeneracy (G)

min over all ordering (Max Left degree)

If we subtract a vertex from the graph then the degree of its adjacent vertex will be affected by (-1)

Planar Graph is 5-degenerate.

K-degenerate graph?
A graph can be destroyed by repeatedly removing a vertex with at most K degree

Tree → Acyclic & connected graph
trees & forest are 1-degenerate.

K-degenerate graph is (K+1)-degenerate also

For finding degeneracy, you don't have to discover all 2^n induced subgraph → we go for n induced subgraphs

Page No.
Date :

⇒ Orientation of an undirected graph : all ~~the~~ edges have direction that is an XOR direction of each edge

⇒ 2^m orientations (2 options for each edge)

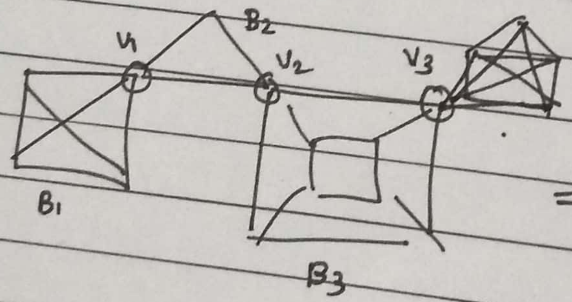
→ Orientation of complete graphs : TOURNAMENTS

→ A block is maximal sub-graph (which is 2 connected) with 0-cut vertex

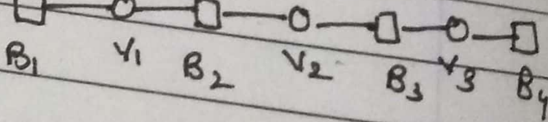
⇒ Every cut-point (vertex) is shared between two blocks

Block cut point tree : Constructed from undirected graph

Block cut point tree leafs will be blocks (can't be cut-vertex as cut vertex is always shared btw two blocks so degree will be even)



⇒ Block cut point tree



We can argue that if we forget triangle & get induced sub-graph B_1 & B_3. ~~The~~ This subgraph has no cut vertex but that doesn't mean its a Block. Block's def^n means subgraph is it should be 2-connected.

⇒ No graph has all vertex as cut vertex

Cut-vertex doesn't mean after removing ~~one~~ it, it gives disconnected graph. It can already be a disconnected graph.

⇒ Def^n is cut-vertex removal increases number of components.

Before I remove the cut-vertex, there is a path for a pair but after, there isn't.

Strongly Connected Component:

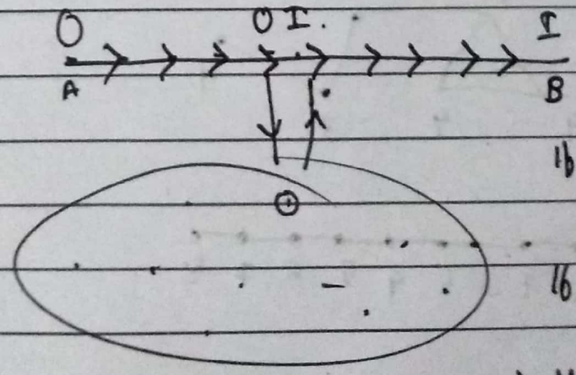
maximal Subgraph with 2-way reachability (directed graph)

⇒ An edge is a cut edge only if doesn't lie on any cycles

⇒ If an undirected graph has a cut-edge, its impossible to give an orientation s.t resulting directed graph is strongly connected graph.

☆ Every Tournaments has an hamiltonial path
↓
a spanning path

Existing path



O → outgoing from path
I → incoming

If A → ~~I~~ or B → O no problem.

If all I or all O no problem

⇒ there must be a OI transformation & we can add this vertex in middle

Every tournament has a triangle.

Kink vertex.

⇒ ∃ a vertex s.t we can be reached in at most 2 hops (TOURNAMENT)

↳ Refer Book Greedy Algorithm

Every tournament has a Kink vertex

⇓

TREES

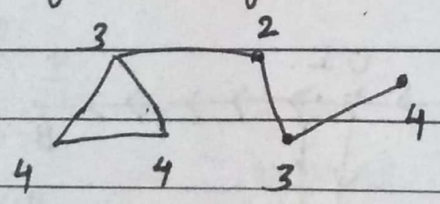
Q ~~Can~~ What are the other graphs apart from complete graph, whose no matter what orientation has kink vertex?

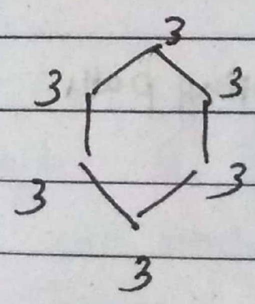
nalization What are the graphs, no matter how we orient it, has a vertex from which we can reach to any vertex in k-hops?

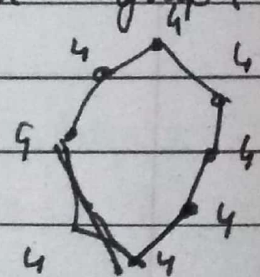
* Distance : length of shortest path.

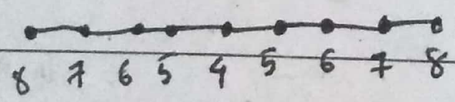
* Eccentricity (Local property, of every vertex):
  Apply BFS ~~for~~ from that vertex
  Get distance array
  Maximum element of d-array
  is eccentricity of the source.

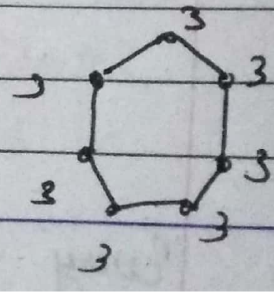
- Same for every vertex in vertex transitive graph











Eccentricity: $\max_{u \in V} d(u, v)$

Radius of the graph: Min eccentricity

Diameter of the graph: Max eccentricity.

Triple optimisation

Rad: min max path Ecc

max Dia

Center: Vertex with min eccentricity

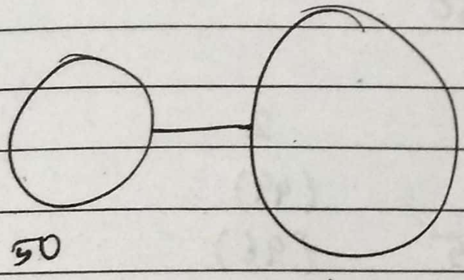
* $(U, V) \in E(G)$.

then eccentricity $|e(u) - e(v)| \leq 1$

Ratio of radius & diameter atmost 2:1

No graph possible with min ecc 10 & max eccentricity 25.

Ex!



50

100

max eccentricity : 76

min eccentricity : 50

adjacent edges distance = 1

so diff btw u & v if distance (u, v) = 1

eccentricity diff = 1

⇒ dist(u, v) = d, then eccentricity diff = d

Hence, diameter can be atmost 2.

⇒ At least two vertices of highest eccentricity

→ Shortest path is induced so

Eccentricity should increase & decrease monotonousl

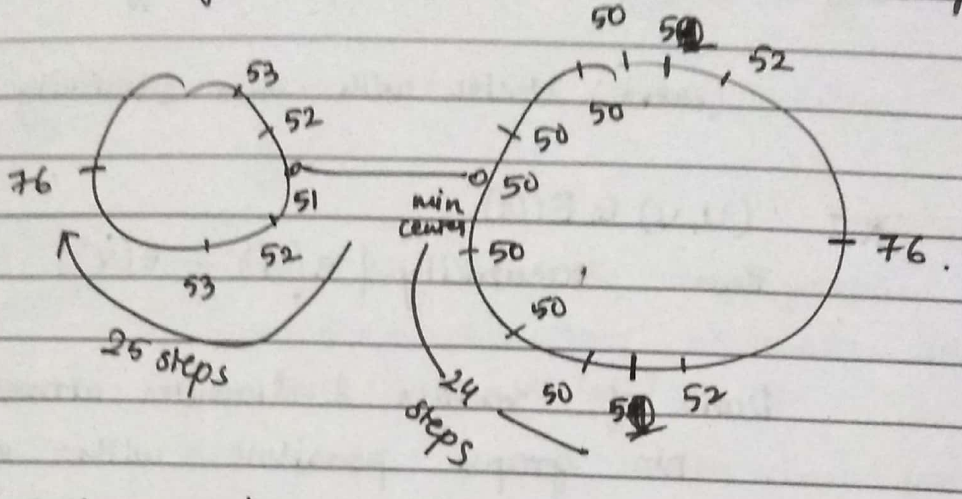
8 7 6 5 4 5 6 7 8

Eccentricity sequence

Page No.
Date :

⇒ Eccentricity more than Minimum has to appear twice.

Tree : Every pair of vertices has a unique path



⇒ any vertex with 24 hops can reach centre than eccentricity will be 50

2 with e 76 — 2.
49 with e 50 — (49)
4 each with e 52 – 75. — (96)
3 with e 51 — 3

150.

→ Max cannot be more than twice of min
Min cannot be less than half the max

element with e 40.
than max can be 80
Min can be 20
Cannot happen simultaneously

Date :

⇒ In super graph, eccentricity of a vertex can be same or decrease but can't increase

Page No.
Date :

1. Take a node & apply BFS.

Caterpillar is a category of trees where if we remove vertices, we get a Path (??)

→ In a rooted tree where there is no leaves, next level contains more nodes than current level

wrt our proof, all leaves are either on odd levels or even levels

non leave nodes

$$|l_{n+1}| \geq |l_n''|$$

We wanna argue

$$|l_1'| + |l_3'| + |l_5'| - - - - |l_{2k+1}'| > |l_0'| + |l_2'| + - - - - |l_{2k}'|$$

$l_1 = l_1' \cup l_1''$

leaf nodes of level$_1$; non leaf nodes of level$_1$

$\Rightarrow$ $|l_1'| + |l_3'| + - - . |l_{2k+1}'| + |l_1''| + |l_2''| + - - |l_{2k+1}'| > |l_0| + |l_2| + - - |l_{2k}|$

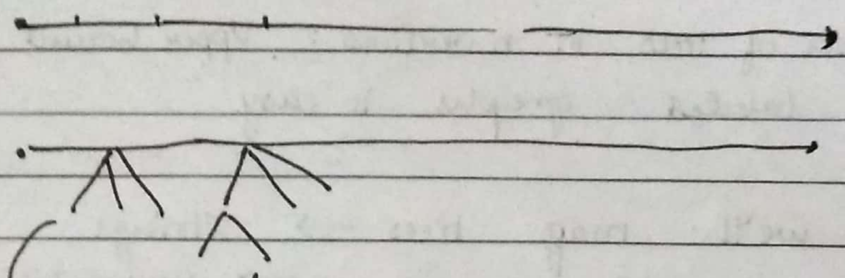
Page No.
Date :

→ Eccentricity of nodes in the longest path remains same even if we remove ~~leaves~~ branches

→ Only a vertex or two vertices (an edge) is the centre of the graph.

Step 1

Step 2



only one depth as main branch is longest as assumed

level 2 maximum

→ Eccentricity of vertices outside the longest path have eccentricity at least one more that vertex nearest to it on the path

→ Any tree with K edges can be embedden on a graph (G) whose vertices have at least K degree.

not just induced subgraphing

so connection here means connection at graph G

connection in graph G doesn't mean connection here

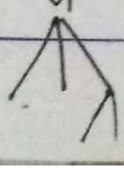
take an edge

v_1

v_2

if all K edges have v_1 vertex as one vertex, we can map any vertex of G to v_1 & it has degree at least K → so np.

v_1



x edges with vertex v_1

x degree

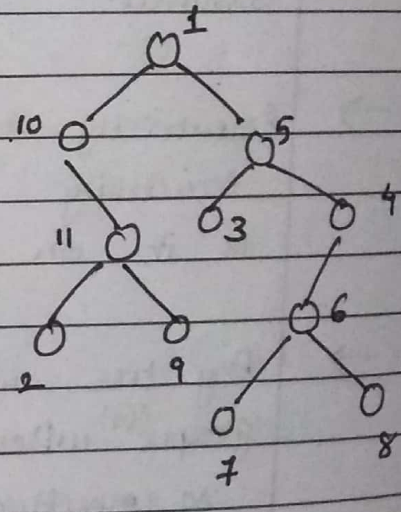
Minimum degree K.

⇒ No. of distinct trees on n vertices

fingerprinting of trees.

Number of trees in n vertices : Upper bound : $2^{n/2}$
Counting labeled graphs is easy

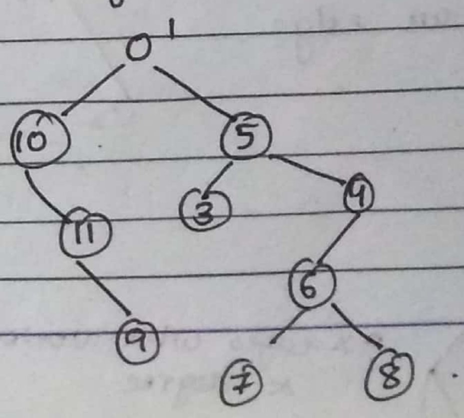
⇒ to count we'll map trees → strings
string alphabets → vertex labels.
string length to represent a unique tree → n-2.
⇒ number of trees in n vertices = n^{n-2}.



h: {2, 9, 3, 7, 8}

Find lowest index leaf & write down its unique neighbour
↓ Delete that leaf node.
Can have migration from internal node to leaf.
→ Do it till 2 nodes left.

⇒ 2 lowest idx → neighbour = 11



11 → new tree

new lowest idx leaf = 3
so 5

11 5

11 5 6 8 → now migration happens

6 from internal node to leaf.

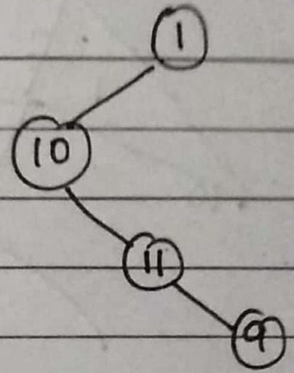
11 5 6 6 4 5 1 ~~11~~ ~~10~~

↓

new tree

leaf with least idx is $\boxed{1}$.

So neighbour is 10.



11 5 6 6 4 5 1 10 11.

Reconstruct original tree from this

perty of this code.

=> algo will not terminate prematurely

↳ tree with at least two nodes has at least two leaves (Refer Theorem)

Number of times a node appears is Degree − 1

at every step degree reduced by 2.

* neighbour of leafs degree decreased with 1

so if its removed at some point, it has to become leaf at some point, so has to go from degree to 1 (at at step degree reduces by one of neighbour of a leaf)

So a node appears degree − 1 times.

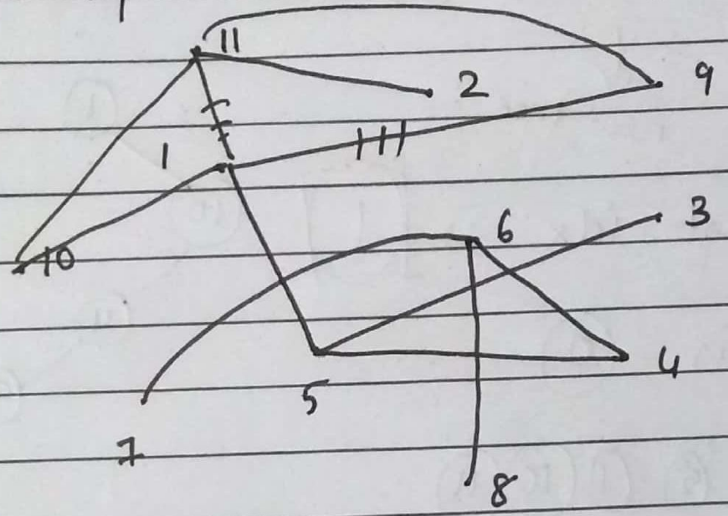
Scanned by CamScanner

Reconstruction

First step in Algo: length of string +2 = n.

$\therefore$ $\boxed{n = 11}$

Sec step : Draw out nodes ; we'll add edges later



Smallest idx not appearing in code → leaf.

Here 2. symbol is 11.

$\therefore$ 2 – 11 edge.

1 is in code
2 dissapears from vertex set.
Now 3 is missing

remove 11 from code

5 6 6 4 5 1 10 11

Now smallest idx not appearing is 3

So next deleted node must be 3

symbol 5 ⇒ 3-5 edge.

↓

6 6 4 5 1 10 11

Now smallest idx not appear is 7.

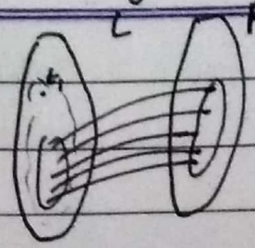
So 7 – 6 → edge

6 4 5 1 10 11

Next is 8

Proof: If Halls Cond^n holds true ⇒ Perfect Matching

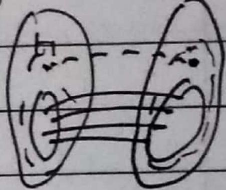
Augmenting Path.



Assuming: Largest matching
& Hall's cond^n holds true.

~~to~~ Now cardinality of matched subset in L & R is same according to def^n of matching

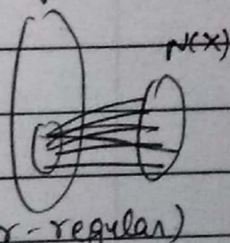
Take t_1 in the group (We'll find a new group of
Case 1: The new neighbours according to Hall's cond^n)
neighbour add^n element is neighbour of t_1 ie
No problem with this.



Case 2: There is no new add^n to neighbour set.

Corollary: Every r-Regular Bipartite graph has Perfect Matching
unless its zero regular
Assume Hall's cond^n is false ⇒ ∃ $x \subseteq |L|$ where $|N(x)| \ngeq |X|$
total number of edges going outside $X = r.|X|$

~~∀x⊆|L|~~ Neighbour of X N(X) Ki average degree



$$\frac{r.|X|}{|N(x)|} > r$$

But every Vertex in N(X) has at most r degree (r-regular)
So Contradiction

→ Removing perfect matching from r-regular graph gives r-1 regular graph

~~Every~~ Graph with Δ max degree will req Δ matchings

Bipartite Graph Perfect Matching

Halls theorem

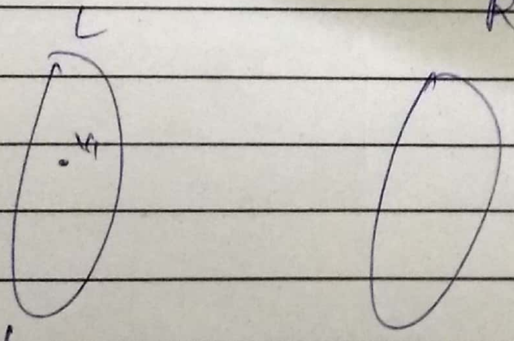
Without loss of generality, assume graph is connected

Necessary condⁿ: $|L| = |R|$

★ Halls ~~Con~~ ~~Theorem~~ Condition:

If there's no perfect matching, then there's matching saturating the smallest partite set

For every ~~subset~~ subset of L, the cardinality of the ~~neighbour set of subset~~ subset



L ~~saturated~~.

→ Degree of ~~of~~ $v_1 \in [1, |R|]$

→ ~~Degree~~ a Number of neighbours of v_1 & v_2 at least 2

If 1 then ~~one~~ one of them match to neighbours & other will be left ⇒ not saturated

Hall's Condition: $\forall x \subseteq L,\ |N(X)| \geq |X|$

Hall's theorem: A graph has perfect matching if & only if Halls Conditions holds

4 5 1 10 11

Date : 1 ~~2~~ ~~3~~ 4 5 ~~6~~ ~~7~~ ~~8~~ 9 10 11

⇒ Next is 6 ⟶ 6 not appearing

so 6 − 4 edge.

5 1 10 11

~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~ 11

Next is 4.

4 − 5

1 10 11

Next is 5.

5 − 1

10 − 11

Next is 1.

1 − 10

11

Next is 9

9 − 10

10 11

Next is 1.

1 − 10.

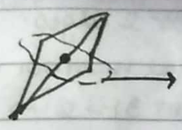
Remaining 11.

Cuts & Connectivity

→ Cut vertex → Vertex cut
→ Cut edge. → Edge cut

- Vertex Connectivity
- Edge Connectivity
- K-(vertex)-connected
- K-edge connected
- Local connection
- Global connection

→ Cut-vertex : Removal of vertex <u>increases</u> the no. of connected components (not just disconnected) ie graph can be disconnected.

 → This graph doesn't have cut-vertex but vertex cut of size 3.

→ Vertex cut : Set of vertices on whose removal gives us <u>disconnected</u> component

→ Cut edge : A single edge whose removal <u>increases</u> the no. of connected components. Increase is of 1.

So if graph is already disconnected, vertex cut size = 0

Graph is K-connected means it cannot be disconnected with less that K vertices

Graph is K edge connected means it cannot be disconnected with ~~re~~moval of less than K edge.

Size of ~~the~~ smallest set of edge which disconnects the graph is edge connectivity

$\lambda_{x,y}$ → local connectivity
Number of ~~edges~~ vertex to remove to disconnected x & y.

$K_G = \min \lambda_{x,y} \quad \forall x \, \forall y$.
↓
Global connectivity

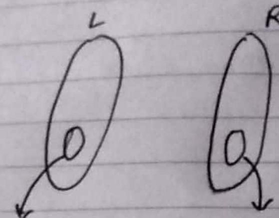
★ ★ Halls theorem , Berges Theorem

16/10/17

For all graph $\beta \geq \alpha'$

Two partitions → Vertex cover And Bipartite Graph

L R



Minimum cardinality vertex cover

Vertex cover in Left   Vertex Cover in Right

To prove: For bipartite graph, $\beta = \alpha'$
as $\beta \geq \alpha'$, we need to prove $\beta \leq \alpha'$

Proof: We will find a matching of cardinality $= |VC_L| + |VC_R|$

We'll match VC_L to Non-VC_R & vice-versa

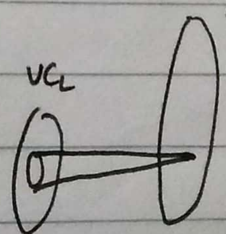
$$\beta = \cancel{(L, R)}\ (VC_L, VC_R)$$
$$B_1 = (VC_L,\ \text{Non-}VC_R)$$
$$B_2 = (\text{Non } VC_L,\ VC_R)$$

We won't have edge with (Non VC_L, Non VC_R) because its a vertex cover and this edge, if both end points are outside, VC won't cover this edge, which is meaningless

→ We are going to demonstrate Hall's condn for VCL in B1.

Non-VC_R

VC_L



Suppose Halls condn fails
ie two vertices have one common neighbour only

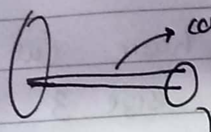
But here vertex cover which is minimum has contradiction. We can cover both edges with one vertex in Non-VCR. So can minimilize further
Contradiction

Thus assumption wrong
Hall's condⁿ true for VC_L.

Matching at least of VC size
$\beta \geq \alpha'$ $\Rightarrow \beta = \alpha'$

Corollary of Tutte's theorem: Any cubic (3-regular) graph without a cut-edge has a perfect matching
Peterson's Corollary to Tutte's theorem

Take an arbitrary vertex set V



t

↓ at most 3t edges crossing

can't have only one otherwise cut-edge.

→ odd component

For 2 edges.
total number of degree is $3t-2$ → odd
not possible.

∀ odd component there can't be 0, 1, 2 edges
so we'll have exactly 3 edges.

Tutte's condition is true ⇒ Perfect Matching