◦ Compute deg seq of complement graph.

29/8.

→ isomorphic

necessary cond. : deg seq of graph = deg seq of complement

- deg of vertices of a self complementing graph = $\frac{n-1}{2}$

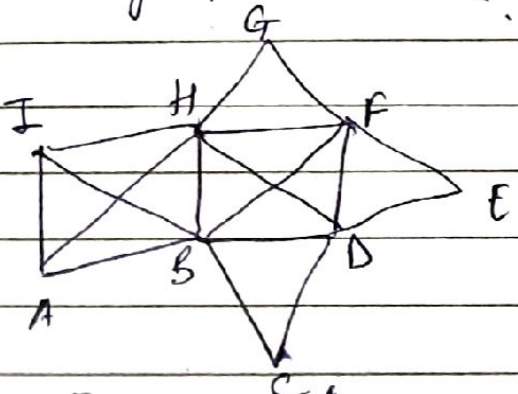
Bipartite graphs

- special case of n-partite graphs ~~where~~ with $n=2$.
- a graph whose vertices can be divided into atmost 2 incident vertex set.

- complete subgraph / clique and independent set are complement of one another.

- Independent set is a subset of vertices s.t induced subgraph has 0 edges b/w them.

Ex



- A single vertex is a clique
- max (IS ∩ clique) = 1

$\alpha \rightarrow$ largest IS $\Rightarrow$ NP hard problem.
ω ~~⊗~~ $\rightarrow$ " clique $\rightsquigarrow$ polynomial time

$\alpha = 4$ ω ~~⊗~~ $= 4$.

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

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

from $\Rightarrow$ ~~⊗~~ $n(A) + n(B) = n(A \cup B) + n(A \cap B)$

• every vertex is a independent set as well as clique.

BG ~~⊗~~ • edgeless graph is the only 1 partite graph graph.

~~partite~~

• BG has valid bi-partition
– show the division of vertices for a bipartite graph (simply stating that a graph is bipartite is not enough)

~~⊗ Imp~~: Property of BP Graphs:
→ every connected bipartite graph, there is a unique way of partitioning (bipartition)
(~~Not the~~

Q: Supergraph, subgraph.

Graph	Operation	Result
Bipartite	Subgraph	BP
Bipartite	supergraph	~~[illegible]~~
Not BP	Subgraph	~~[illegible]~~
Not BP	supergraph	Not BP

1) BP is closed under subgraph operation
4) NBP is closed under supergraph operations
→ contrapositive statements

⇒ A graph is BP if & only if it does not have an odd length cycle.

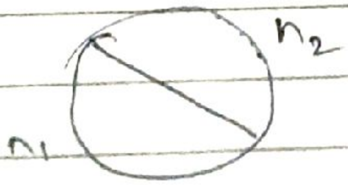
(Ex. to show that a graph is not BP
— subgraph has odd cycle.)

tree edges, cross edges → BFS.

→ Run BFS   if cross edge : not BP

3/9 ~~[illegible]~~ → if there is an odd cycles, then there has to be an induced odd cycle.

→ induce cycle ⇒ w/ no chords.



n_1 & n_2 are integers
$n_1 + n_2$ ⇒ odd

3/09/19

BFS
↓
classify edges as tree + cross edge

vertical horizontal
↳ Not bipartite

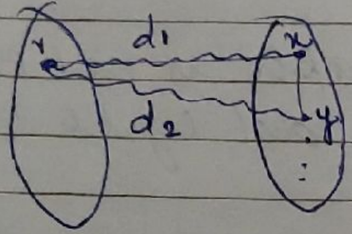
Algorithm:

correctness proved by contraction

1) Run BFS
2) for each $(x, y) = e \in E(G)$
3) if $\pi(x) \neq y$ and $\pi(y) \neq x$ ⇒ tree edge
4) then if $d[x] = d[y]$ then
5) the graph is not bipartite (return NB)
6) return bipartite

$O(N+E)$ time

→ Assume that one of the sets is not independent



d_1 and d_2 are odd

$d_1 + d_2 + xy$ = odd
→ closed walk ~~of odd length~~ (odd)
every closed walk has an odd cycle.
contradiction,

Q.

→ odd level vertices + even level vertices in separate sets.

- finding the largest subgraph to make a graph bipartite is NP hard problem
↳ cannot be solved efficiently.

⇒ divide vertices into 2 sets + ignore/remove local edges
↓
equal # in each set

→ if vertices ~~has more~~ is in the larger set,
during migration → # edges ↑'s.
do this until no such vertex exists

$d_G(v) = d_L(v) + d_F(v)$
local foreign

$\boxed{\Sigma d_G(v) = 2 \cdot |E|}$

$d_F(v) \geq \frac{d_G(v)}{2}$

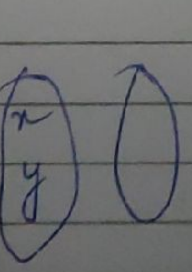
local degree of bipartitite subgraph $= \Sigma d_F(v)$
$\geq \frac{\Sigma d_G(v)}{2}$ $|E|$

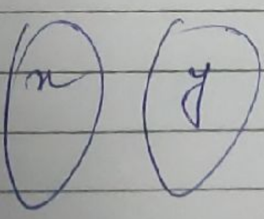
- $2|E_B| = \frac{1}{2} \cdot 2 \cdot |E|$

* approximation algo for bipartition
in the set
→ place the vertex where # neighbours is less.

$d_G(x) = d_L(x) + d_A(x)$

⇒ every connected bipartite graph has a unique ~~of~~ bipartition
→ Prove by contradiction





d_{xy} = even d_{xy} = odd