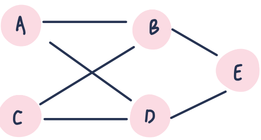
GRAPH

CH 10.1 DEC. 19. 20

NOTES BY: LISA :)



VERTEX : 5

EDGES : 6 ~ ex set of edge = {A, B}

- ↳ UNDIRECTED {A,B} ORDER DONT MATTER A - B
- ↳ DIRECTED (A,B) ORDER MATTERS A → B

INDEGREE : HOW MANY EDGES POINT AT THIS NODE

OUT DEGREE : HOW MANY EDGES POINT OUT OF THIS NODE

THIS GRAPH IS REPRESENTED AS:

({A,B,C,D,E}, {{A,B}, {A,D}, {B,C}, {B,E}, {C,D}, {D,E}})

VERTICES OF THE GRAPH

THE SET OF EDGES

SIMPLE GRAPH

- UNDIRECTED EDGES
- NO PARALLEL EDGES (NOT ∥ EDGES)
- NO LOOPS (EDGE TO ITSELF)

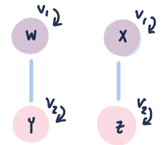
A SIMPLE GRAPH G = (V, E)

IS BIPARTITE IF:

EACH VERTEX IS IN EXACTLY 1 OF V_1 OR V_2,

+ EVERY EDGE IS BETWEEN A VERTEX IN V_1 + A VERTEX IN V_2

EXAMPLE

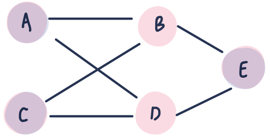


✓ each vertex is in exactly V_1 or V_2

✓ each edge is between V_1 or V_2

V_1 : {W, X}

V_2 : {Y, Z}



✓ ✓ BOTH CONDITIONS

V_1 : {A, C, E}

V_2 : {B, D}

~ every edge connects vertices in different groups ~

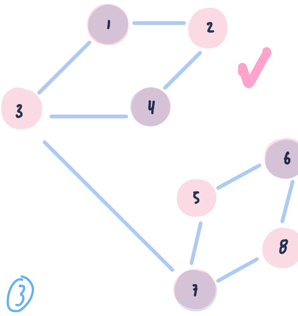
APPLICATION:

- COMPUTER NETWORKS / COMPUTER → PRINTERS

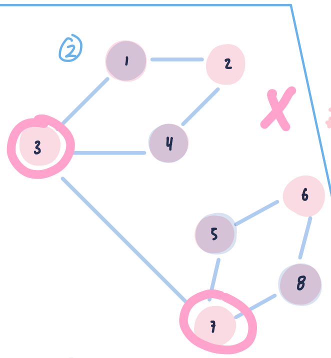
↳ WOULDNT HAVE COMPUTER → COMPUTER OR PRINTER → PRINTER

MORE EXAMPLES:

①



②



V_1 : {1, 4, 6, 7}

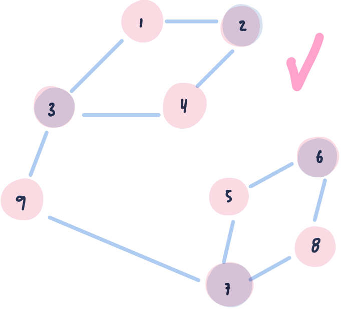
V_2 : {2, 3, 5, 8}

V_1 : {1, 4, 5, 8}

V_2 : {2, 3, 6, 7}

CANT BE SAME GROUP

③



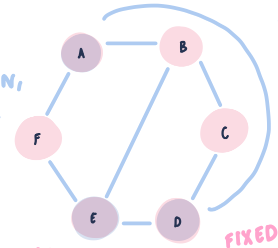
V_1 : {2, 3, 6, 7}

V_2 : {1, 4, 5, 8, 9}

MORE EXAMPLES

DISPLAYING AN INVALID BIPARTITION, DOES NOT MEAN THE GRAPH IS NOT BIPARTITE.

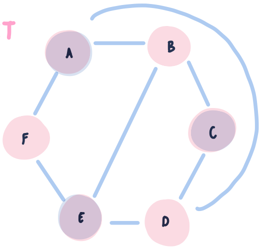
ALWAYS VERIFY 4 YOURSELF.



IS BIPARTITE.

SOMEONE JUST COLORED IT WEIRD DOES NOT MEAN IT IS NOT A BIPARTITE GRAPH

FIXED IT



V_1 : {A, C, E}

V_2 : {B, D, F}

PROVING BIPARTITION:

LET'S ASK QUESTIONS:

- WHAT ABOUT THIS CYCLE MAKES THE GRAPH NOT BIPARTITE?

- THIS IS AN ODD LENGTH CYCLE.

$v_1, v_2, \ldots v_k$ (for some odd k)

WE PROVE NO ODD LENGTH CYCLE IS BIPARTITE.

THE PROOF

ASSUME FOR CONTRADICTION FOR GRAPH G IS BIPARTITE:

THEN THERE IS SOME BIPARITION (V_1, V_2)

ASSUME WITHOUT LOSS OF GENERALITY v_1 IS IN SET V_1.

v_1 IS IN SET V_1, THERE ARE 2 POSSIBILITIES → $v_1 \in V_1$ or $v_1 \in V_2$

SAME PROOF FOR v_1 in V_1 as v_1 in V_2

↳ (IF $v_1 \in V_2$, WE RENAME V_1 to V_2)

(OTHERWISE, IT MUST BE IN V_2, SO WE CAN SWAP V_1 + V_2)

INDUCTION

v_2 MUST BE $\in V_2$, since it shares an edge w/ v_1 + $v_1 \in V_1$

v_3 MUST BE $\in V_1$, since it shares an edge w/ v_2 + $v_2 \in V_2$

EACH ODD INDEXED VERTEX IS IN V_1 + EACH EVEN INDEXED VERTEX IS IN V_2

INDUCTIVELY, THE PREVIOUS VERTEX IS IN THE OTHER SET.

SO v_1 + v_k $\in V_1$ (both odd indexed)

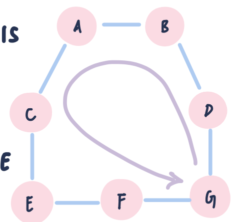
THEY SHARE AN EDGE, CANT BOTH BE IN V_1

↳ CONTRADICTION = NOT BIPARTITE

PROVEN BY CONTRADICTION

SHOWING PROOF

WE USE THIS RING TO SHOW NOT BIPARTITE

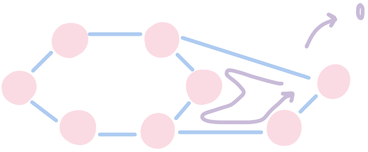


CAREFUL!

YOU CAN HAVE BAD JUSTIFICATIONS.

JUST BE CAUSE YOUR JUSTIFYING SOMETHING, DOESNT MEAN IT IS JUSTIFIED

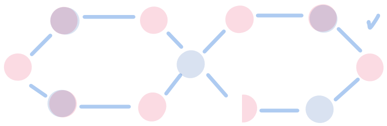
MORE EXAMPLE :



ODD CYCLE DETECTED

NOT BIPARTITE.

ODD CYCLE



✓ BIPARTITE

EVEN CYCLE

CONNECTED :

THERE IS A PATH BETWEEN ANY TWO VERTICES

↑ IF A GRAPH w/ AN ODD LENGTH CYCLE IS NOT BIPARTITE, THEN

IF A GRAPH HAS NO ODD LENGTH CYCLE, MUST IT BE BIPARTITE ?

LET G be a (connected) simple graph, assume G has no odd length cycle.

→ (we will show G is bipartite by exhibiting a bipartition)

we're looking for graphs that are not bipartite + do NOT have an odd length cycle.

CHOOSE AN ARBITRARY VERTEX ; CALL IT V_0

IF THE SHORTEST PATH FROM v_i to V_0 IS ODD, PUT v_i INTO V_1

IF THE SHORTEST PATH FROM v_i to V_0 IS EVEN, PUT v_i INTO V_2 (V_0 will itself be in V_2 ; distance 0)

= BIPARTITE

** NEED TO SHOW THERE IS NO EDGE BETWEEN TWO VERTICES IN THE SAME GROUP **

SUPPOSE THERE IS NO EDGE BETWEEN TWO VERTICES IN V_1 : v_i + v_j

THEN, WE'RE SAYING THE PATH FROM v_i TO V_0 IS ODD + THE PATH FROM v_j → V_0 IS ODD

SO THEN THE EDGE CONNECTING v_i TO v_j CANNOT BE THERE, SINCE THE 2 VERTICES IN THE SAME GROUP CANNOT BE CONNECTED

odd path from V_0 to v_i : length $2k_1 + 1$ for some k_1 int

odd path from v_j to V_0 : length $2k_2 + 1$ for some k_2 int

$v_i \to v_j$: length 1

whole cycle length : $(2k_1+1) + (2k_2+1) + 1$ = ODD

Suppose same thing for V_2, same process but with even cycle lengths:

produces $(2k_1) + (2k_2) + 1$ → still odd cycle length.

CONTRADICTION: THERE ARE NO ODD LENGTH BIPARTITE CYCLES

↳ THE CYCLE LENGTH IS ODD MEANING THAT SOMETHING IN V_1 IS CONNECTED TO SOMETHING IN V_2

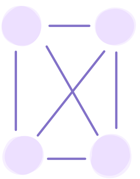
SOME MORE TERMINOLOGY :

COMPLETE GRAPH : K_n

K_n

A GRAPH WITH n VERTICES WHERE EVERY VERTEX IS HAS AN EDGE TO EVERY OTHER VERTEX

K_4 :



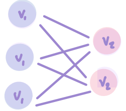
ALSO. NOT BIPARTITE. HAS AT LEAST 1 ODD LENGTH CYCLE

COMPLETE BIPARTITE GRAPH : $K_{n,m}$

$K_{n,m}$

A BIPARTITE GRAPH W/ A GROUP OF n + ANOTHER GROUP OF m VERTICES.

$K_{3,2}$:

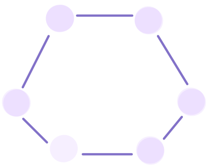


CYCLE GRAPH : C_n

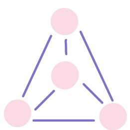
C_n

CYCLE OF LENGTH n HAS n VERTICES CONNECTED IN A SIMPLE GRAPH

C_6 :



CAREFUL : A GRAPH CAN BE MULTIPLE AT ONCE



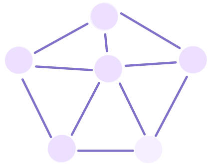
THIS IS K_4 + W_3

WHEEL GRAPH : W_n

W_n

ONE COPY OF C_n + AN ADDITIONAL VERTEX THAT CONNECTS TO ALL OTHER VERTEX.

W_5 :



ICS 21 ~ CH 10.3

ISOMORPHISM

CONDITIONS 4 ALL ISOMORPHIC GRAPHS.

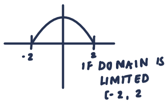
WE HAVE 2 GRAPHS:

- G = (V, E)
- H = (W, F)

+ THEY ARE **ISOMORPHIC** IFF THERE IS A FUNCTION $f: \overset{\text{DOMAIN}}{V} \rightarrow \overset{\text{CODOMAIN}}{W}$ SUCH THAT

① IT IS **ONE-TO-ONE** + **ONTO**

FUNCTION W/ THESE PROPERTIES:

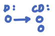


SUCH THAT **ONE-TO-ONE**: EVERY ELEMENT IN THE CODOMAIN IS MAPPED AT MOST ONCE

↳ CANNOT HAVE

ONTO: EVERY ELEMENT IN THE CODOMAIN IS MAPPED AT LEAST ONCE

↳ CANNOT HAVE



SUCH THAT FOR ANY (DIFFERENT) VERTICES a, b in V

{a, b} IS AN EDGE IN E iff {f(a), f(b)} is an edge in F

② ★ G + H 'S **VERTEX + EDGE SETS NEED TO BE SAME SIZE**

ELSE, EDGE IN ONE GRAPH WILL MATCH WITH NON-EDGE IN OTHER.

③ ★ MUST HAVE **SAME DEGREE SEQUENCE**

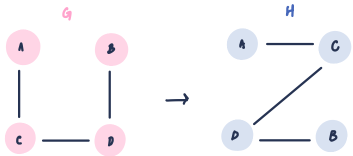
↳ TO PERSERVE EDGES

IN ADDITION TO COND 3:

★ **CYCLE SIZES MATCHES BUT NEED TO PROVE IT**

PROVING THIS IS HARD. CANNOT JUST PROVE THAT THEY HAVE CYCLES OF THE SAME SIZE BUT HAS TO PROVE THAT NO CYCLE IS NOT THE SAME SIZE AS WELL.

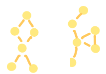
ARE THESE GRAPHS ISOMORPHIC?



· looks like scrambled pink graph

- ✓ BOTH ONTO
- ✓ BOTH ONE-TO-ONE
- ✓ G + H HAS THE SAME EDGES

EX:



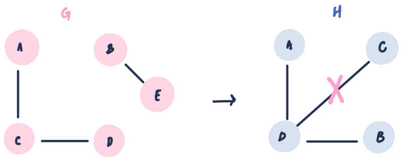
4 2 2 2 1 1

NOT ISOMORPHIC

· ONE HAS CYCLE ONE DOES NOT

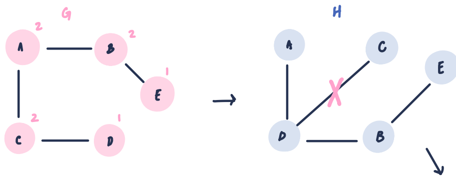
BUT

JUST BECAUSE IT SATISFIES ALL 3 CONDITIONS DOES NOT MEAN THEY ARE ISOMORPHIC



X NO SAME SIZE VERTEX SET

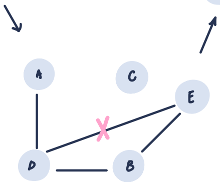
INTRODUCE ANOTHER VERTEX



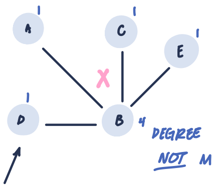
DEGREE SEQUENCE: 2 2 2 1 1

X NO SAME # OF EDGES

X LACK CYCLE



NOW THERE IS CYCLE.
BUT STILL NO SAME EDGE #



DEGREE SEQUENCE DOES NOT MATCH

4 1 1 1

PROVE ISOMORPHIC:

PROVE AN ISOMORPHISM

(CANNOT JUST SHOW THAT ISOMORPHIC INVARIANTS HOLD)

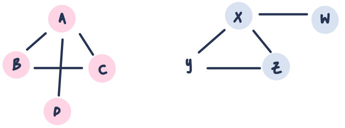
PROVE NOT ISOMORPHIC:

ARGUE IN GENERAL NO ISOMORPHISM FAIL

(CAN SHOW AN ISOMORPHIC INVARIANT FAILS)

BASICALLY MEANING 2 GRAPHS HAVE THE EXACT SAME PROPERTIES

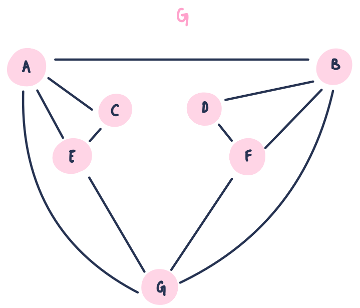
ISOMORPHIC



f(a) = x
f(b) = y
f(c) = z
f(d) = w

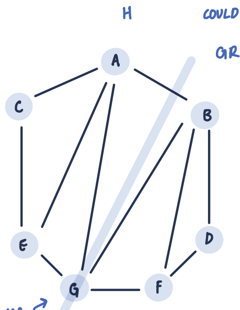
LEC 22

ISOMORPHIC OR NO ?



4 4 4 3 3 2 2

- ✓ # VERTICES
- ✓ # EDGES
- ✓ SAME EDGE SEQUENCE
- ✓ ONE-TO-ONE
- ✓ ONTO



COULD REFLECT GRAPH ALONG THIS LINE & STILL WORKS

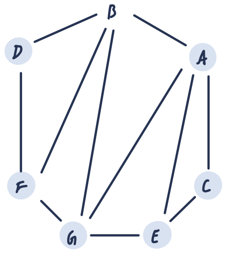
YES, ISOMORPHIC

THIS ONE HAS TO BE G, BECAUSE IT HAS THE TWO DEGREE 3 NODES AS NEIGHBORS

4 4 3 3 2 2

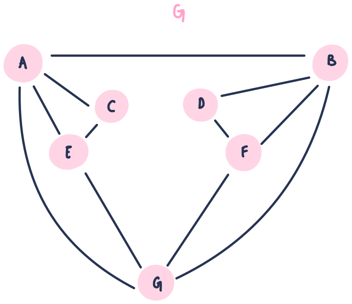
TO CONSTRUCT H, WE LOOK FOR UNIQUE FEATURES OF G AND MATCH THEM TO THE VERTICES ON G

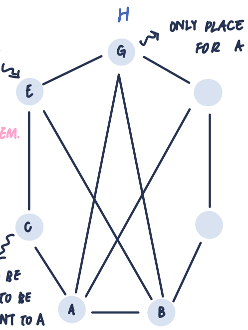
OTHER FORM



STILL ISOMORPHIC ✓

ISOMORPHIC OR NO ?





DEG. 3 ADJ. TO C MUST BE HERE

BUT! THIS IS A PROBLEM.

ONLY PLACE G CAN BE FOR A VALID SOLUTION

HAS TO BE HERE TO BE ADJACENT TO A

★ AT EACH STEP, IF THERE IS ISOMORPHISM, THEN ____ MUST MAP TO _____

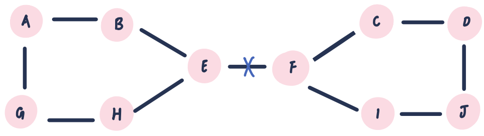
↳ IN THIS CASE. IF THERE IS ISOMORPHISM AT E, THEN E MUST MAP TO A, C, G, BUT IT CANNOT, SO IT IS NOT ISOMORPHIC

CANNOT SIMPLY ASSERT # OF LENGTH K CYCLES

~ NEED TO PROVE IT ~

★ EXAMINE EACH POSSIBLE CASE AT EACH STEP TO CHECK FOR ISOMORPHISM

CONNECTIVITY



IN THIS GRAPH, IS THERE ANY EDGE WE COULD REMOVE, HAVING DONE SO, THE GRAPH BECOMES DISCONNECTED.

↳ YES, SINGLE EDGES TO DISCONNECT GRAPH : EDGE {E, F}

☆ A **CUT EDGE**

↳ YES, A GROUP/SET OF EDGES TO REMOVE : {EDGE {B, E}, EDGE {G, H}}

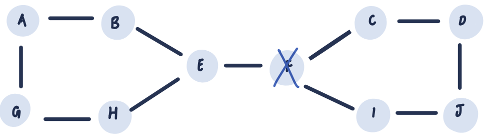
☆ AN **EDGE CUT** (MORE IMPORTANT)

EDGE CONNECTIVITY of a SIMPLE GRAPH :

" size of smallest edge cut

↳ denoted as $\lambda(G)$ for edge connectivity of G

ex: edge connectivity $\lambda(G)$ of 1 = removing a single edge, disconnects the graph.



IN THIS GRAPH, IS THERE ANY VERTEX WE COULD REMOVE, HAVING DONE SO, THE GRAPH BECOMES DISCONNECTED.

↳ YES, SINGLE VERTEX TO DISCONNECT GRAPH : F

☆ A **CUT VERTEX**

↳ YES, A GROUP/SET OF VERTICES TO REMOVE : {B, H}

☆ AN **VERTEX CUT**

VERTEX CONNECTIVITY of a SIMPLE GRAPH :

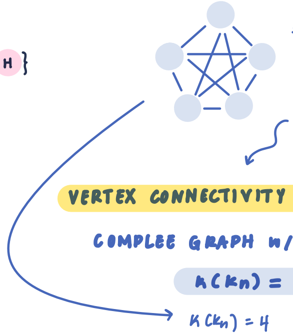
" size of smallest edge cut

↳ denoted as $\kappa(G)$ for vertex connectivity of G

KAPPA

ex: VERTEX connectivity $\kappa(G)$ of 1 = removing a single VERTEX, disconnects the graph.

THERE'S A SPECIAL CASE!



NO VERTEX CUT!

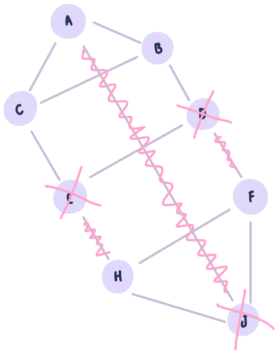
↳ IT IS A COMPLETE GRAPH, WILL HAVE TO REMOVE ALL OF THE VERTICES.

VERTEX CONNECTIVITY ON GRAPH K_n,

COMPLEE GRAPH W/ n VERTICES,

$\kappa(K_n) = n - 1$

$\kappa(K_n) = 4$



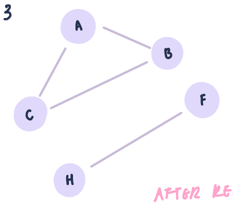
EDGE CONNECTIVITY?

3 (EXAMPLE)

REMOVING {E, H}, {D, F}, {A, J}

VERTEX CONNECTIVITY?

3



AFTER REMOVING E, D, J

mindeg (G) >= $\lambda(G)$

THERE'S ALWAYS AN EDGE CUT OF SIZE MIN DEG(G)

↳ CUT EVERY EDGE ADJACENT TO THE VERTEX W/ MIN DEGEE

= DISCONNECTS GRAPH.

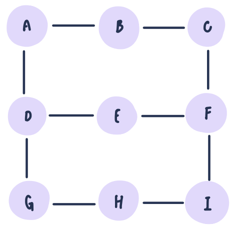
☆ REMOVING A or J WOULD NOT WORK, THE REST OF THE GRAPH IS STILL CONNECT. GRAPH IS NOT SPILT INTO SUBGRAPHS

↗ AFTER REMOVAL, STILL A FULLY CONNECTED GRAPH MINUS A OR J

☆ **GOAL OF VERTEX CUT** → REMOVE A SET OF VERTICES THAT RESULTS IN TWO OR MORE SUBGRAPHS

LEC 23

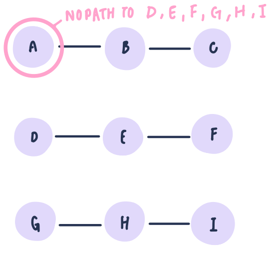
CONNECTED COMPONENTS



WE SAY A SIMPLE GRAPH IS **CONNECTED** IF FOR ANY TWO VERTEXES, THERE'S A PATH BETWEEN THEM

← LIKE IN THIS GRAPH

WHAT IF WE MODIFIED THE GRAPH A LITTLE:



IS THIS GRAPH CONNECTED?

NO, BUT IT HAS **CONNECTED COMPONENTS**

↳ C.C. OF A SIMPLE GRAPH = MAXIMAL SETS OF VERTEXES IN WHICH THERE IS A PATH BETWEEN ANY TWO VERTEX IN THE SET

MAXIMAL: CANNOT ADD ANOTHER COMPONENT W/O BREAKING THE PROPERTIES

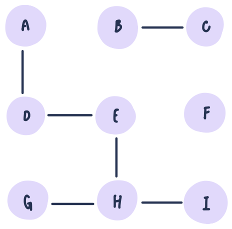
EVERY VERTEX IS IN
EXACTLY ONE
CONNECT COMPONENT

THE CONNECTED COMPONENTS HERE:

{A, B, C}
{D, E, F}
{G, H, I}

WHY IS {A, B} NOT A CONNECTED COMPONENT?

↳ NOT MAXIMAL, C NEEDS TO BE ADDED



THE CONNECTED COMPONENTS:

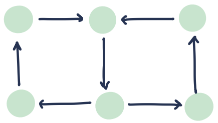
{F} → HAS ITS OWN CONNECTED COMPONENT
{A, D, E, G, H, I}
{B, C}

CONNECTIVITY: DIRECTED GRAPHS

FOR SIMPLE DIRECTED GRAPHS

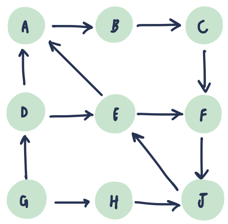
STRONGLY CONNECTED IF & ONLY IF

THERE IS A DIRECTED PATH FROM EACH VERTEX TO EACH OTHER VERTEX



WEAKLY CONNECTED IF N ONLY IF

ONLY THE UNDERLYING UNDIRECTED GRAPH IS CONNECTED



DAG: DIRECTED ACYCLIC GRAPH

DIRECTED GRAPH W/ NO SIMPLE CYCLE

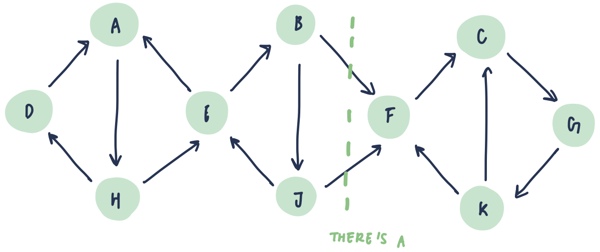
STRONGLY CONNECTED COMPONENTS

{A, B, C, E, F, J}

{H}

{G}

{D}



THERE'S A CUT HERE NOTHING ON ↷ SIDE CAN GET TO THE OTHER SIDE

NOT STRONGY CONNECTED.

HAS STRONGLY CONNECTED COMPONENTS

{A, B, D, E, H, J}

{C, F, G, K}

N-CUBE: A SPECIAL GRAPH

Q_n

Q_0 = single vertex

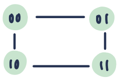
Q_n = consists of 2 copies of of Q_{n-1}

WITH EDGES (UNDIRECTED) BETWEEN THE 2 COPIES' CORRESPONDING COMPONENTS

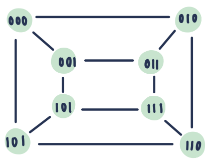
Q_0:

Q_1:

Q_2:



Q_3:



THERE'S AN EDGE IF THEY DIFFER BY EXACTLY ONE BIT.

LET a_n BE THE # OF VERTICES IN Q_n

LET b_n BE THE # OF EDGES IN Q_n

CAN WE CONSTRUCT A RECURSIVE FORMULA FOR # OF VERTICES, a_n?

$a_0 = 1$ $\quad a_n = 2 * a_{n-1}$ (Q_n is built from 2 copies of Q_{n-1})

↓

closed form:

$a_n = 2^n$

CAN WE CONSTRUCT A RECURSIVE FORMULA FOR # OF EDGES, b_n?

$b_0 = 0$ (no edges yet) $\quad b_n = 2(b_{n-1}) + a_{n-1} \rightarrow b_n = 2(b_{n-1}) + 2^{n-1}$

↑ solve for closed form solution