1.6 PATHS & CYCLES

DEFN.

A WALK IN A GRAPH G IS A FINITE SEQUENCE OF VERTICES AND EDGES OF THE FORM

$$v_0 e_1 v_1 e_2 v_2 \cdots e_{k-1} v_{k-1} e_k v_k$$

WHERE EDGE e_i JOINS VERTEX v_{i-1} TO v_i FOR $1 \le i \le k$. (ALSO: $v_0 - v_k$ WALK.)

IF G IS SIMPLE ITS SUFFICIENT TO JUST LIST THE VERTICES OF A WALK.

NOTE THE VERTICES AND EDGES OF A WALK NEED NOT BE DISTINCT.

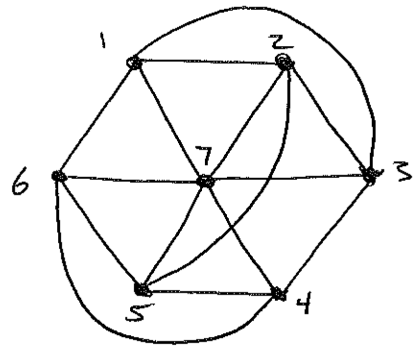
A WALK IS CLOSED IF THE INITIAL AND TERMINAL VERTICES ARE IDENTICAL: $v_0 = v_k$.

A WALK IN WHICH ALL EDGES ARE DISTINCT IS CALLED A TRAIL.

A TRAIL IN WHICH ALL VERTICES ARE DISTINCT (EXCEPT POSSIBLY $v_0 = v_k$) IS CALLED A PATH.

A CLOSED PATH IS CALLED A CYCLE.

EX.



- A WALK WHICH IS NOT A TRAIL:
  1 7 6 1 7 2 3 4 5 2 7 1
- A TRAIL WHICH IS NOT A PATH:
  1 7 4 6 7 3
- A PATH: 3 7 6 5 2 1
- A CYCLE: 1 2 3 4 5 6 1

NOTE THAT GIVEN $u, v \in V(G)$ ANY u-v WALK CAN, BY DELETING VERTICES AND EDGES, BE REPLACED BY A SUBSEQUENCE WHICH IS A u-v PATH.

EX. $W = 1\,7\,6\,1\,7\,2\,3\,4$
$P = 1\,7\,2\,3\,4$

DEFN.
WE SAY $u, v \in V(G)$ ARE <u>CONNECTED</u> IF G CONTAINS A u-v PATH.

G IS ITSELF CALLED <u>CONNECTED</u> IF EVERY PAIR OF ITS VERTICES ARE CONNECTED. OTHERWISE G IS <u>DISCONNECTED</u>.

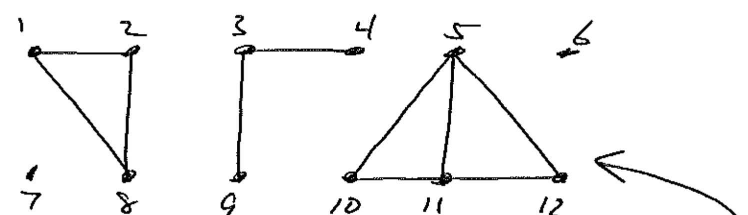
GIVEN $u \in V(G)$ DEFINE

$$C(u) = \{ v \in V(G) \mid u \text{ is CONNECTED TO } v \}.$$

THEN $G[C(u)]$, THE SUBGRAPH OF G INDUCED BY $C(u)$, IS CALLED THE <u>CONNECTED COMPONENT CONTAINING u</u>.

NOTE THE COMPONENT OF G CONTAINING u IS A CONNECTED SUBGRAPH OF G. IN FACT THE COMPONENTS OF G ARE PRECISELY THOSE SUBGRAPHS WHICH ARE MAXIMAL WITH RESPECT TO THE PROPERTY OF BEING CONNECTED.

<u>EX.</u>
G



$\omega(G) = 5$

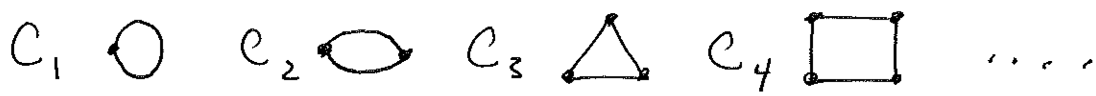
$C(5) = \{5, 10, 11, 12\}$. $G[C(5)]$

WE DENOTE BY $\omega(G)$ THE NUMBER OF CONNECTED COMPONENTS IN G.

OBVIOUSLY G IS CONNECTED IF AND ONLY IF $\omega(G) = 1$.

THE <u>LENGTH</u> OF A PATH (OR CYCLE) IS ITS NUMBER OF EDGES. A CYCLE OF LENGTH K IS SOMETIMES CALLED A K-CYCLE.

UP TO ISOMORPHISM THERE IS JUST ONE GRAPH CONSISTING OF A SINGLE K-CYCLE, DENOTED C_K.



SIMILARLY P_K DENOTES THE GRAPH CONSISTING OF A SINGLE PATH (NON CLOSED) OF LENGTH K.

<u>THEOREM</u>

LET G BE A GRAPH WITH AT LEAST TWO VERTICES. THEN G IS BIPARTITE IF AND ONLY IF G CONTAINS NO ODD LENGTH CYCLES.

<u>PROOF:</u>

($\Rightarrow$) SUPPOSE G IS BIPARTITE WITH BIPARTITION $V(G) = X \cup Y$. LET $C = v_0 v_1 v_2 \cdots v_{K-1} v_K$ BE A K-CYCLE IN G. (i.e. $v_0 = v_K$.)

WE MUST SHOW THAT k IS EVEN.

SUPPOSE $v_0 \in X$. (THE ARGUMENT IS SIMILAR IF $v_0 \in Y$ AND WE OMIT IT.) THEN $v_1 \in Y$, $v_2 \in X$, $v_3 \in Y$, ... etc..

EVIDENTLY $v_i \in X$ IFF i IS EVEN. BUT

$$v_k = v_0 \in X$$

SO k MUST BE EVEN.

($\Leftarrow$) SUPPOSE G CONTAINS NO ODD LENGTH CYCLES.

WE MAY ASSUME G IS CONNECTED, FOR SUPPOSE THE RESULT IS PROVED FOR CONNECTED GRAPHS. IF G IS DISCONNECTED AND HAS NO ODD LENGTH CYCLES, THEN EACH OF ITS COMPONENTS HAS NO ODD CYCLES. G IS THEN THE UNION OF BIPARTITE COMPONENTS, AND AS SUCH IS ITSELF BIPARTITE.

GIVEN $u, v \in V(G)$ DEFINE

$$d(u,v) = \text{LENGTH OF A SHORTEST } u\text{-}v \text{ PATH}$$

SUCH A PATH MUST EXIST SINCE G IS NOW ASSUMED TO BE CONNECTED, SO $d(u,v)$ IS WELL DEFINED.

FIX A VERTEX $s \in V(G)$ (CALLED SOURCE)
AND SET

$$X = \{u \in V(G) \mid d(s,u) \text{ is even}\}$$

$$Y = \{u \in V(G) \mid d(s,u) \text{ is odd}\}$$

THEN clearly $X \cap Y = \phi$ AND $X \cup Y = V(G)$.

<u>CLAIM</u>: IF G CONTAINS AN EDGE uv WITH BOTH $u \in X$ AND $v \in X$ OR BOTH $u \in Y$ AND $v \in Y$, THEN G CONTAINS AN ODD LENGTH CYCLE.

THE RESULT FOLLOWS FOR SINCE G HAS NO ODD CYCLES, NO SUCH EDGE EXISTS, WHENCE X, Y IS A BIPARTITION IN G.

<u>PROOF OF CLAIM</u>:
SUPPOSE $uv \in E(G)$ WITH $u \in X$ AND $v \in X$.
(THE CASE $u \in Y$, $v \in Y$ is similar.)

LET P BE A SHORTEST s-u PATH AND Q BE A SHORTEST s-v PATH

$$P: s \leadsto z \leadsto u$$

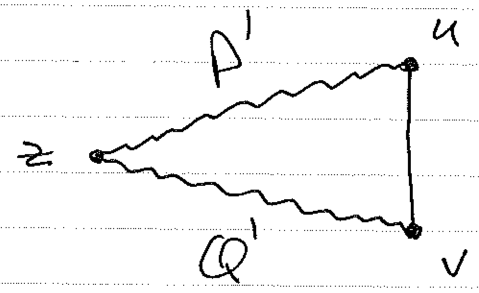
$$Q: s \leadsto z \leadsto v$$

LET Z BE THE LAST (i.e. FURTHEST FROM S) VERTEX THAT P & Q HAVE IN COMMON. THEN THE SEGMENTS

P' : $z \rightsquigarrow u$

Q' : $z \rightsquigarrow v$

HAVE THE SAME PARITY, (i.e. BOTH even OR BOTH ODD.) THUS



is AN ODD LENGTH cycle in G, AS REQUIRED.

///