Lecture 03
Eulerian Circuits,
Vertex Degrees and Counting

Dr. Rajiv Misra
Associate Professor
Dept. of Computer Science & Engg.
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Preface

Recap of previous Lecture:

- In the previous lecture, we have discussed the useful properties of connection, paths, and cycles and how the statements in graph theory can be proved using the principle of induction.

Content of this Lecture:

- In this lecture, we will discuss eulerian circuits, the fundamental parameters of a graph i.e. the degree of the vertices, counting and extremal problems.

Eulerian Circuits

- A graph is ***Eulerian*** if it has a closed trail containing all edges.
- We call a closed trail a ***circuit*** when we do not specify the first vertex but keep the list in cyclic order.
- An ***Eulerian circuit*** or ***Eulerian trail*** in a graph is a circuit or trail containing all the edges.

Even Graph, Even Vertex

- An ***even graph*** is a graph with vertex degrees all even.

- A vertex is ***odd*** [***even***] when its degree is odd [even].

Contd…

- Our discussion of Eulerian circuits applies also to graphs with loops; we extend the notion of vertex degree to graphs with loops by letting each loop contribute 2 to the degree of its vertex.

- This does not change the parity of the degree, and the presence of a loop does not affect whether a graph has an Eulerian circuit unless it is a loop in a component with one vertex.

Maximal Path

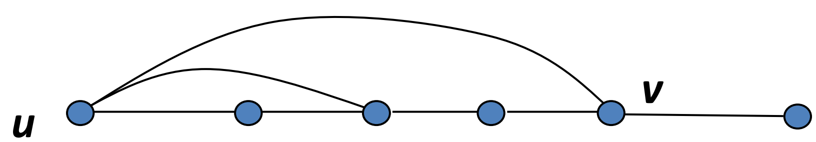
- A ***maximal path*** in a graph ***G*** is a path ***P*** in ***G*** that is not contained in a longer path.

  - When a graph is finite, no path can extend forever, so maximal (non-extendible) paths exist.

Lemma: If every vertex of graph G has degree at least 2, then G contains a cycle. 1.2.25

Proof:

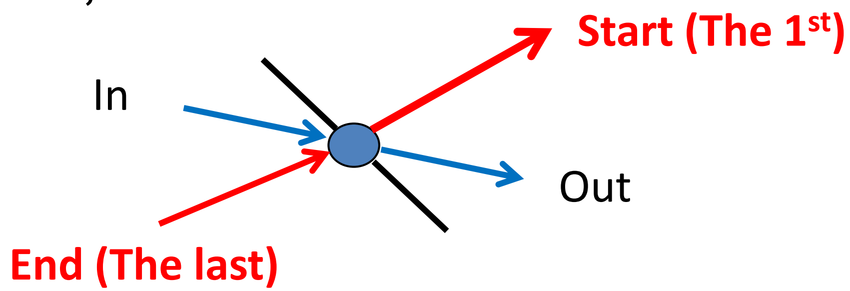
- Let ***P*** be a maximal path in ***G***, and let ***u*** be an endpoint of ***P***
- Since ***P*** cannot be extended, every neighbor of ***u*** must already be a vertex of ***P***
- Since ***u*** has degree at least 2, it has a neighbor ***v*** in ***V*** **(*P*)** via an edge not in ***P***
- The edge ***uv*** completes a cycle with the portion of ***P*** from ***v*** to ***u***



Theorem: A graph *G* is Eulerian if and only if it has at most one nontrivial component and its vertices all have even degree. 1.2.26

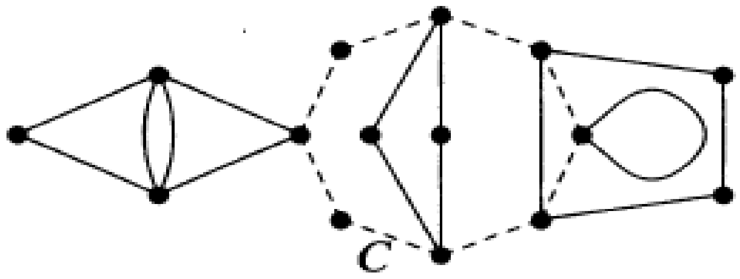
Proof: (*Necessity*) *G* is Eulerian only if it has at most one nontrivial component and its vertices all have even degree (G is Eulerian ⇒ its vertices have even degree …)

- Suppose that ***G*** has an Eulerian circuit ***C***
- Each passage of ***C*** through a vertex uses two incident edges, and the first edge is paired with the last at the first vertex. Hence every vertex has even degree.
- Also, two edges can be in the same trail only when they lie in the same component, so there is at most one nontrivial component.



Proof: (*Sufficiency*) G is Eulerian if it has at most one nontrivial component and its vertices all have even degree

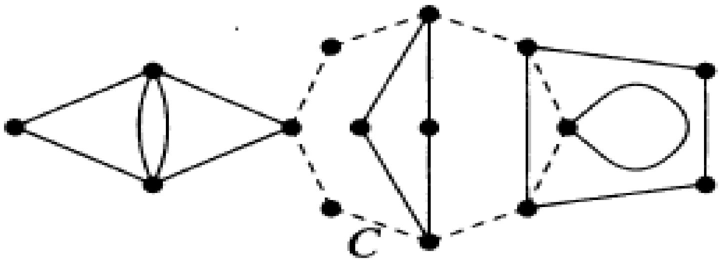
- Assuming that the condition holds, we obtain an Eulerian circuit using induction on the number of edges, m
- Basis step: $m = 0$. A closed trail consisting of one vertex suffices



Proof: (*Sufficiency*)

- Induction step: $m > 0$.
  - With even degrees, each vertex in the nontrivial component of G has degree at least 2.
  - By Lemma 1.2.25, the nontrivial component has a cycle C.
  - Let G' be the graph obtained from G by deleting $E(C)$.
  - Since C has 0 or 2 edges at each vertex, each component of G' is also an even graph.
  - Since each component is also connected and has fewer than m edges, we can apply the induction hypothesis to conclude that each component of G' has an Eulerian circuit.

Proof: (*Sufficiency*)(continued)



- To combine these into an Eulerian circuit of *G*, we traverse *C*, but when a component of *G'* is entered for the first time we detour along an Eulerian circuit of that component.
- This circuit ends at the vertex where we began the detour.
- When we complete the traversal of *C*, we have completed an Eulerian circuit of *G*.

TONCAS

- In the characterization of Eulerian circuits, the necessity of the condition is easy to see. This also holds for the characterization of bipartite graphs by absence of odd cycles and for many other characterizations.

- Nash-Williams and others popularized a mnemonic for such theorems: **TONCAS**, meaning “The Obvious Necessary Conditions are Also Sufficient”.

Extremality

- The proof of Lemma 1.2.25 is an example of an important technique of proof in graph theory that we call **extremality**.
- When considering structures of a given type, choosing an example that is extreme in some sense may yield useful additional information.
- For example, since a maximal path *P* cannot be extended, we obtain the extra information that every neighbor of an endpoint of *P* belongs to *V(P)*

Proposition: If G is a simple graph in which every vertex has degree at least k, then G contains a path of length at least k. If $k \geq 2$, then G also contains a cycle of length at least $k+1$. 1.2.28

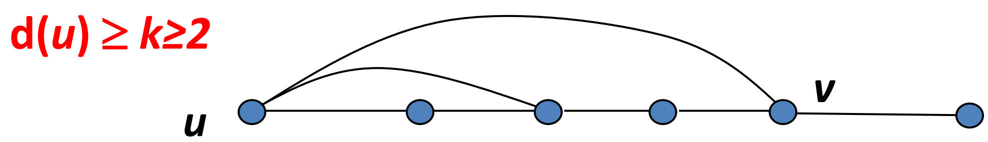
Proof: (1/2)

- Let u be an endpoint of a maximal path P in G.
- Since P does not extend, every neighbor of u is in $V(P)$.
- Since u has at least k neighbors and G is simple, P therefore has at least k vertices other than u and has length at least k.

Proposition: If G is a simple graph in which every vertex has degree at least k, then G contains a path of length at least k. If $k \geq 2$, then G also contains a cycle of length at least $k+1$. 1.2.28

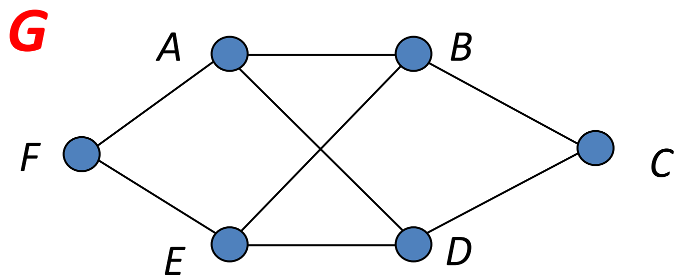
Proof: (2/2) Contd..

- If $k \geq 2$, then the edge from u to its farthest neighbor v along P completes a sufficiently long cycle with the portion of P from v to u.



Degree

- The **degree** of vertex v in a graph G, written $d_G(v)$, or $d(v)$, is the number of edges incident to v, except that each loop at v counts twice
- The **maximal degree** is $\Delta(G)$
- The **minimum degree** is $\delta(G)$

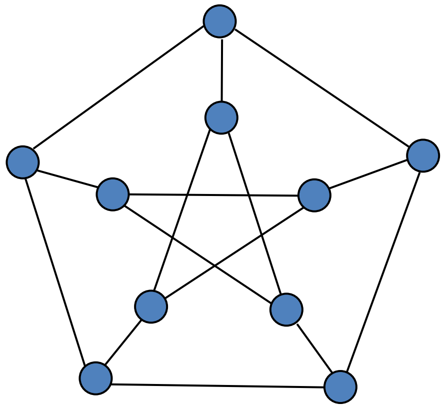


$d(B) = 3$, $d(C) = 2$

$\Delta(G) = 3$, $\delta(G) = 2$

Regular

- ***G*** is ***regular*** if $\Delta(G) = \delta(G)$
- ***G*** is ***k-regular*** if the common degree is ***k***.
- The ***neighborhood*** of ***v***, written $N_G(v)$ or $N(v)$ is the set of vertices adjacent to ***v***.



3-regular

Order and size

- The ***order*** of a graph ***G***, written ***n (G),*** is the number of vertices in ***G***.
- An ***n-vertex graph*** is a graph of order *n*.
- The ***size*** of a graph ***G***, written ***e*** (***G***), is the number of edges in *G*.
- For ***n*∈*N***, the notation [***n***] indicates the set {***1,…, n*** }.

Proposition: (Degree-Sum Formula)– Handshaking lemma

If G is a graph, then $\sum_{v \in V(G)} d(v) = 2e(G)$ 1.3.3

Proof:

- Summing the degrees **counts each edge twice**,
  - since each edge has two ends and contributes to the degree at each endpoint.

$\blacksquare$

- **Corollary** 1.3.4. In a graph G, the average vertex degree is $\frac{2e(G)}{n(G)}$,and hence $\delta(G) \leq \frac{2e(G)}{n(G)} \leq \Delta(G)$.

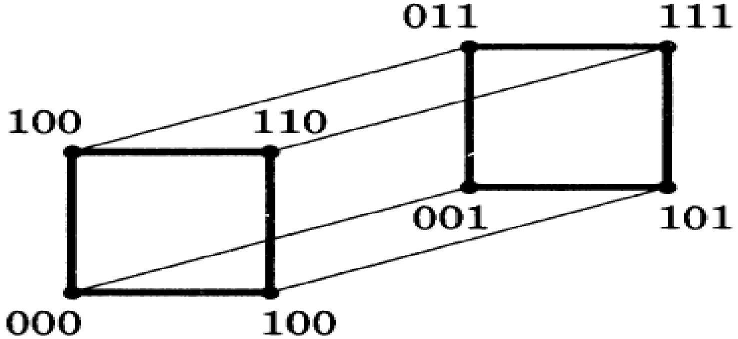
- **Corollary** 1.3.5. Every graph has an even number of vertices of odd degree. No graph of odd order is regular with odd degree.

- **Corollary** 1.3.6. A *k*-regular graph with *n* vertices has *nk/2* edges.

K-dimensional cube or hypercube

Definition:

- The k-**dimensional cube** or **hypercube** Q_k is the simple graph whose vertices are the k-tuples with entries in {0, 1} and whose edges are the pairs of k-tuples that differ in exactly one position. A j-**dimensional subcube** of Q_k is a subgraph of Q_k isomorphic to Q_j

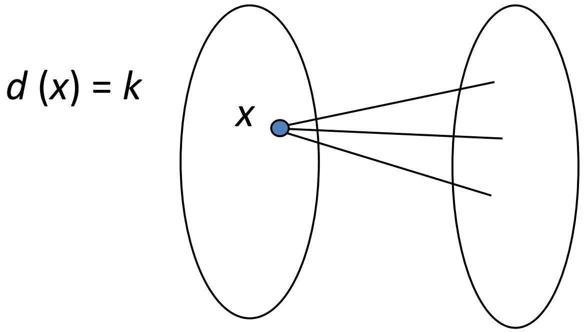


- Above we show Q_3 . The hypercube is a natural computer architecture. Processors can communicate directly if they correspond to adjacent vertices in Q_k . The k-tuples that name the vertices serve as addresses for the processors.

Theorem: If $k>0$, then a k-regular bipartite graph has the same number of vertices in each partite set. 1.3.9

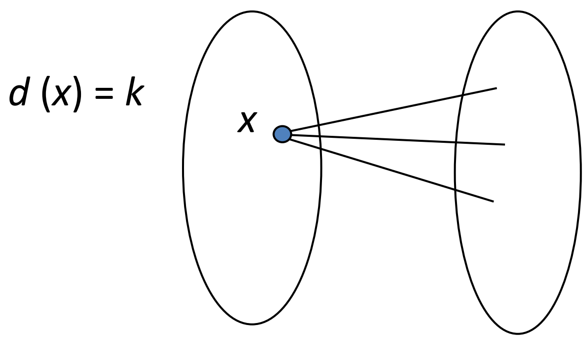
Proof:

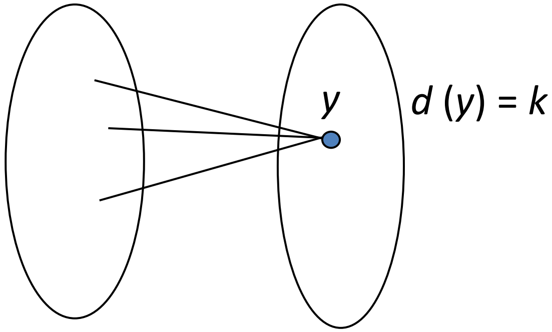
- Let G be a k-regular X,Y - bigraph.
- Counting the edges according to their endpoints in X yields $e(G) = k\,|X|$.



Proof:

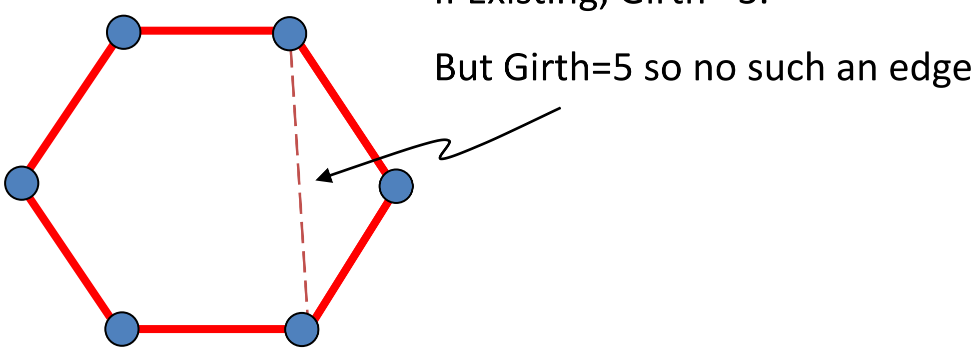
- Counting them by their endpoints in Y yields $e(G)=k|Y|$
- Thus $k|X| = k|Y|$, which yields $|X|=|Y|$ when $k>0$





A technique for counting a set 1/3 1.3.10

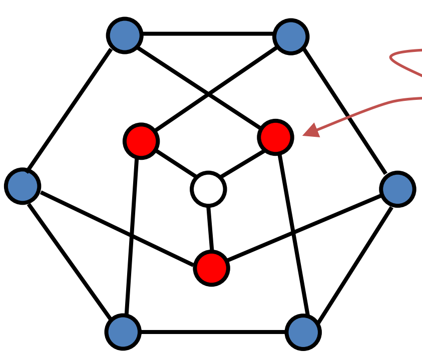
- Example: The Petersen graph has ten 6-cycles
  - Let G be the Petersen graph.
  - Being 3-regular, G has ten copies of $K_{1,3}$ (claws). We establish a one-to-one correspondence between the 6-cycles and the claws.
  - Since G has girth 5, every 6-cycle F is an induced subgraph.
    - see below
  - Each vertex of F has one neighbor outside F.
    - $d(v)= 3$, $v \in V(G)$



– Since nonadjacent vertices have exactly one common neighbor (Proposition 1.1.38), opposite vertices on *F* have a common neighbor outside *F*.

– Since *G* is 3-regular, the resulting three vertices outside *F* are distinct.

– Thus deleting *V(F)* leaves a subgraph with three vertices of degree 1 and one vertex of degree 3; it is a claw.

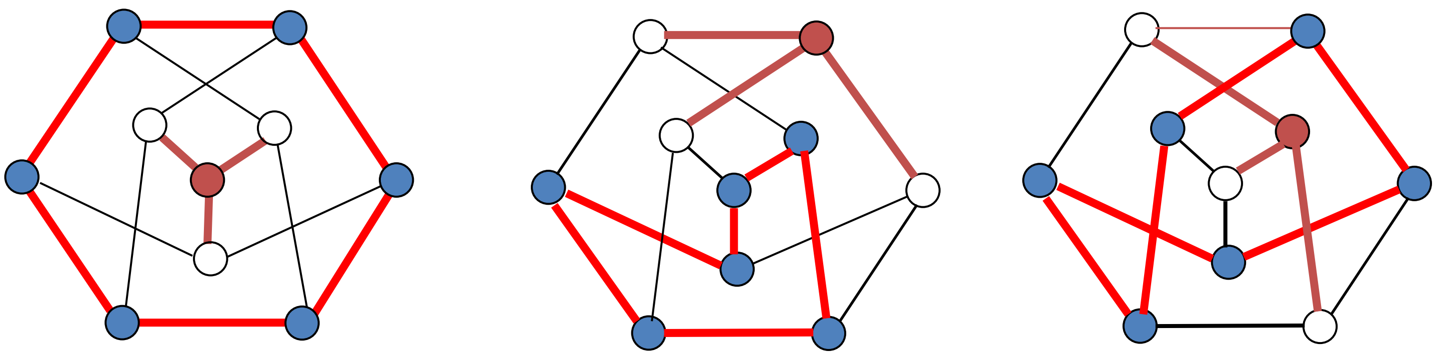


Common neighbor of opposite vertices

If the neighbors are not distinct, $d(v)>3$

A technique for counting a set 3/3 1.3.10

- It is shown that each claw *H* in *G* arises exactly once in this way.
- Let *S* be the set of vertices with degree 1 in *H*; *S* is an independent set.
- The central vertex of *H* is already a common neighbor, so the six other edges from *S* reach distinct vertices.
- Thus *G - V(H)* is 2-regular. Since *G* has girth 5, *G - V(H)* must be a 6-cycle. This 6-cycle yields *H* when its vertices are deleted. ■



Extremal Problems

- An **extremal problem** asks for the **maximum or minimum value** of a function over a class of objects.

- For example, the maximum number of edges in a simple graph with n vertices is $\binom{n}{2}$

Proposition: The minimum number of edges in a connected graph with *n* vertices is *n*-1. 1.3.13

Proof:

- By proposition 1.2.11, every graph with ***n*** **vertices** and ***k*** **edges** has at least ***n-k*** **components**.
- Hence every ***n*-vertex** graph with fewer than ***n*-1 edges** has at least two components and is disconnected.
- The contrapositive of this is that every connected ***n*-vertex** graph has at least ***n-1*** **edges**. This lower bound is achieved by the **path** P_n. ■

Conclusion

- In this lecture, we have discussed eulerian circuits, fundamental parameters of a graph i.e. vertex degrees, counting and extremal problems.

- In upcoming lectures, we will discuss further on extremal problems, graphic sequences and directed graphs.