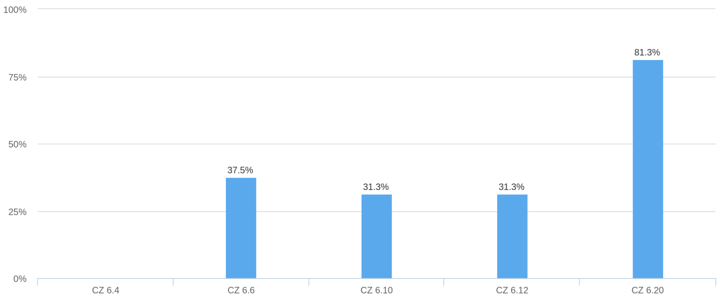
3-10 Traversability

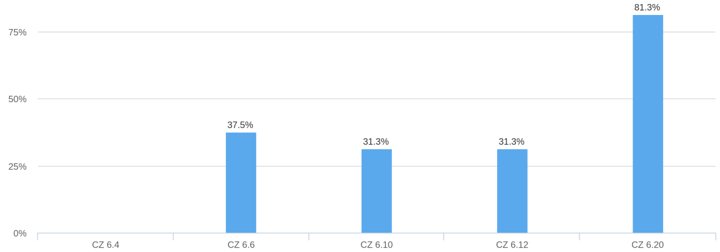
(Part I: Eulerian Graphs)

Hengfeng Wei

hfwei@nju.edu.cn

December 03, 2018


100%
75%
50%
25%
0%
37.5%
31.3%
31.3%
81.3%
CZ 6.4
CZ 6.6
CZ 6.10
CZ 6.12
CZ 6.20



CZ 6.20 (Next Class)

这次习题相对简单【因为是必选的所以随机选了一个】，希望老师可以多回顾一下课本内容，比如哈密尔顿图的各种充分条件和证明，对哈密尔顿图和欧拉图的用法做一些拓展

欧拉图和汉密尔顿图的联系 就是在建模时如何确定图的节点和边

g-cage 对于不同大小而言都是唯一的吗？（书上只给到n=8） Kirkman triple system 和 factorization 的等价推导 Theorem 8.10的证明

6.3，不想看书，看自闭了，希望老师可以直接讲一下

如何打印欧拉回路/欧拉迹/哈密顿回路

定理6.5

无

暂无

无

陶老师上课讲的中国邮递员问题没明白

暂无，我就来抽个奖

none

定理6.12的证明

喵(^･ｪ･^)

希望能讲一下fluery算法

可以总结一下证明的方法，其实每次都可以这样，不一定要课上讲，可以整理之后做成讲义课后发，比如怎么证明有欧拉回路等等

这次习题相对简单【因为是必选的所以随机选了一个】，希望老师可以多回顾一下课本内容，比如哈密尔顿图的各种充分条件和证明，对哈密尔顿图和欧拉图的用法做一些拓展

欧拉图和汉密尔顿图的联系 就是在建模时如何确定图的节点和边

g-cage 对于不同大小而言都是唯一的吗？（书上只给到n=8） Kirkman triple system 和 factorization 的等价推导 Theorem 8.10的证明

6.3，不想看书，看自闭了，希望老师可以直接讲一下

如何打印欧拉回路/欧拉迹/哈密顿回路

定理6.5

无

暂无

无

陶老师上课讲的中国邮递员问题没明白

暂无，我就来抽个奖

none

定理6.12的证明

喵(^•ɪ•^)

希望能讲一下fluery算法🌚

可以总结一下证明的方法，其实每次都可以这样，不一定要课上讲，可以整理之后做成讲义课后发，比如怎么证明有欧拉回路等等

Fleury

这次习题相对简单【因为是必选的所以随机选了一个】，希望老师可以多回顾一下课本内容，比如哈密尔顿图的各种充分条件和证明，对哈密尔顿图和欧拉图的用法做一些拓展

欧拉图和汉密尔顿图的联系 就是在建模时如何确定图的节点和边

g-cage 对于不同大小而言都是唯一的吗？（书上只给到n=8） Kirkman triple system 和 factorization 的等价推导 Theorem 8.10的证明

6.3，不想看书，看自闭了，希望老师可以直接讲一下

如何打印欧拉回路/欧拉迹/哈密顿回路

定理6.5

无

暂无

无

陶老师上课讲的中国邮递员问题没明白

暂无，我就来抽个奖

none

定理6.12的证明

喵(^･ｪ･^)

希望能讲一下fluery算法

可以总结一下证明的方法，其实每次都可以这样，不一定要课上讲，可以整理之后做成讲义课后发，比如怎么证明有欧拉回路等等

Fleury

Chinese Postman Problem (Next Class)

这次习题相对简单【因为是必选的所以随机选了一个】，希望老师可以多回顾一下课本内容，比如哈密尔顿图的各种充分条件和证明，对哈密尔顿图和欧拉图的用法做一些拓展

欧拉图和汉密尔顿图的联系 就是在建模时如何确定图的节点和边

g-cage 对于不同大小而言都是唯一的吗？（书上只给到n=8） Kirkman triple system 和 factorization 的等价推导 Theorem 8.10的证明

6.3，不想看书，看自闭了，希望老师可以直接讲一下

如何打印欧拉回路/欧拉迹/哈密顿回路

定理6.5

无

暂无

无

陶老师上课讲的中国邮递员问题没明白

暂无，我就来抽个奖

none

定理6.12的证明

喵(^•ɪ•^)

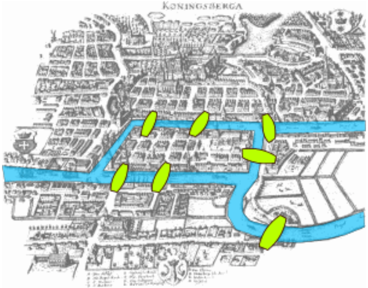
希望能讲一下fluery算法🌚

可以总结一下证明的方法，其实每次都可以这样，不一定要课上讲，可以整理之后做成讲义课后发，比如怎么证明有欧拉回路等等

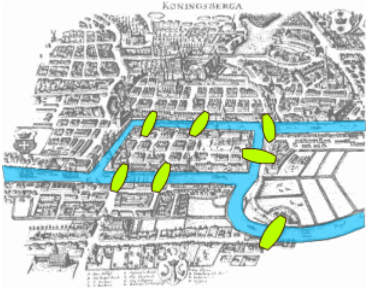
Fleury

Chinese Postman Problem (Next Class)

6.3 Exploration & 6.4 Excursion (Not Required)



Leonhard Euler (1707 – 1783)



Leonhard Euler (1707 – 1783)

Graph Theory Topology


CCTV 14
CNTV
大风车


CCTV 14
CNTV
大风车


田


CCTV 14
CNTV
大风车


田

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873);

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873); (Proofs on PPT from Tao & on CZ).

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873); (Proofs on PPT from Tao & on CZ).

By strong mathematical induction on the number m of edges of G.

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873); (Proofs on PPT from Tao & on CZ).

By strong mathematical induction on the number m of edges of G.

$$\text{Inductive Step} : m = k + 1$$

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873); (Proofs on PPT from Tao & on CZ).

By strong mathematical induction on the number m of edges of G.

$$\text{Inductive Step}: m = k + 1$$

Let $C = u \sim v$ be any trail in G of the maximum length.

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873); (Proofs on PPT from Tao & on CZ).

By strong mathematical induction on the number m of edges of G.

$$\text{Inductive Step} : m = k + 1$$

Let $C = u \sim v$ be any trail in G of the maximum length.

$$v = u$$

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873); (Proofs on PPT from Tao & on CZ).

By strong mathematical induction on the number m of edges of G.

$$\text{Inductive Step} : m = k + 1$$

$$\text{Let } C = u \sim v \text{ be any trail in } G \text{ of the maximum length.}$$

$$v = u \implies C = u \sim u$$

Theorem (Leonhard Euler 1735)

A connected graph G is Eulerian if and only if the degree of each vertex of G is even.

"$\Longleftarrow$" (Carl Hierholzer 1873); (Proofs on PPT from Tao & on CZ).

By strong mathematical induction on the number m of edges of G.

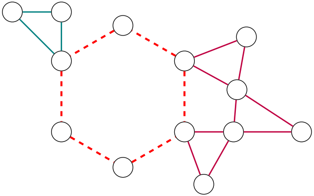
$$\text{Inductive Step} : m = k + 1$$

Let $C = u \sim v$ be any trail in G of the maximum length.

$$v = u \implies C = u \sim u$$

$$H = G - E(C) = \bigcup H_i$$

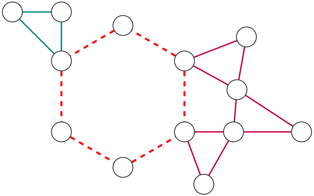
$$H = G - E(C) = \bigcup H_i$$



$$H = G - E(C) = \bigcup H_i$$

(I) $\forall v \in H : \deg(v)$ is even

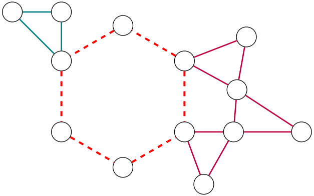
(II) $\forall i : \left| E(H_i) \right| < m$



$$H = G - E(C) = \bigcup H_i$$

(I) $\forall v \in H : \deg(v)$ is even

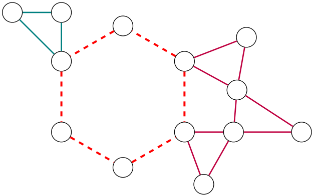
(II) $\forall i : \left| E(H_i) \right| < m$



By I.H., each H_i has an Eulerian circuit C_i.

$$H = G - E(C) = \bigcup H_i$$

$$\text{(I)} \quad \forall v \in H : \deg(v) \text{ is even}$$
$$\text{(II)} \quad \forall i : \Big|E(H_i)\Big| < m$$



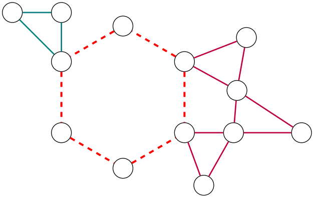
By I.H., each H_i has an Eulerian circuit C_i.

$$\forall i : V(H_i) \cap V(C) \neq \emptyset$$

$$H = G - E(C) = \bigcup H_i$$

(I) $\forall v \in H : \deg(v)$ is even

(II) $\forall i : \left|E(H_i)\right| < m$

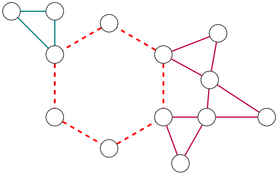


By I.H., each H_i has an Eulerian circuit C_i.

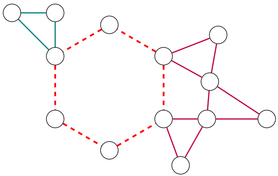
$$\forall i : V(H_i) \cap V(C) \neq \emptyset$$

Glue together each C_i with C to get an Eulerian circuit of G.

```
procedure HIERHOLZER(G)
    u ∈ V(G)
    C ← any circuit u ∼ u in G
    while ∃v ∈ C : deg(v) > 0 do
        H ← G − E(C)
        v ← any vertex in V(C) such that deg(v) > 0
        C′ ← any circuit v ∼ v in H
        C ← C ⊗ C′                              ▷ Glue C′ = v ∼ v with C via v
    return C
```



```
procedure HIERHOLZER(G)
    u ∈ V(G)
    C ← any circuit u ~ u in G
    while ∃v ∈ C : deg(v) > 0 do
        H ← G − E(C)
        v ← any vertex in V(C) such that deg(v) > 0
        C′ ← any circuit v ~ v in H
        C ← C ⊗ C′                      ▷ Glue C′ = v ~ v with C via v
    return C
```



Q : Time Complexity?

procedure HIERHOLZER(G)

  $u \in V(G)$

  $C \leftarrow$ any circuit $u \sim u$ in G

  while $\exists v \in C : \deg(v) > 0$ **do**

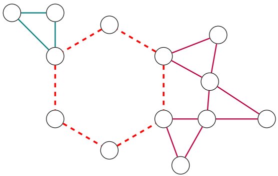
    $H \leftarrow G - E(C)$

    $v \leftarrow$ any vertex in $V(C)$ such that $\deg(v) > 0$

    $C' \leftarrow$ any circuit $v \sim v$ in H

    $C \leftarrow C \otimes C'$ ▷ Glue $C' = v \sim v$ with C via v

  return C



Q : Time Complexity?

Q : Data Structures?

procedure HIERHOLZER(G)

  $u \in V(G)$

  $C \leftarrow$ any circuit $u \sim u$ in G

  while $\exists v \in C : \deg(v) > 0$ **do**

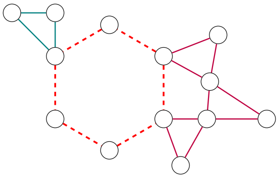
    $H \leftarrow G - E(C)$

    $v \leftarrow$ any vertex in $V(C)$ such that $\deg(v) > 0$

    $C' \leftarrow$ any circuit $v \sim v$ in H

    $C \leftarrow C \otimes C'$ $\triangleright$ Glue $C' = v \sim v$ with C via v

  return C



Q : Time Complexity?

Q : Data Structures?

$O(m)$: Using doubly linked list

Fleury's Algorithm (1883)

(I) $v_0 \in V(G); \quad P_0 = v_0$

(II) Suppose $C_i = v_0 e_1 v_1 e_2 \cdots e_i v_i$.

Choose e_{i+1} from $E(G) - \{e_1, e_2, \cdots, e_i\}$ as follows:

(a) e_{i+1} is incident with v_i

(b) Unless there is no alternative, e_{i+1} is not a bridge of $G - \{e_1, e_2, \cdots, e_i\}$

(III) Stop when step (II) can no longer be implemented

Fleury's Algorithm (1883)

(I) $v_0 \in V(G)$; $\quad P_0 = v_0$

(II) Suppose $C_i = v_0 e_1 v_1 e_2 \cdots e_i v_i$.

Choose e_{i+1} from $E(G) - \{e_1, e_2, \cdots, e_i\}$ as follows:

(a) e_{i+1} is incident with v_i

(b) Unless there is no alternative, e_{i+1} is not a bridge of $G - \{e_1, e_2, \cdots, e_i\}$

(III) Stop when step (II) can no longer be implemented

Fleury's Algorithm (1883)

(I) $v_0 \in V(G)$; $P_0 = v_0$

(II) Suppose $C_i = v_0 e_1 v_1 e_2 \cdots e_i v_i$.

Choose e_{i+1} from $E(G) - \{e_1, e_2, \cdots, e_i\}$ as follows:

(a) e_{i+1} is incident with v_i

(b) Unless there is no alternative, e_{i+1} is not a bridge of $G - \{e_1, e_2, \cdots, e_i\}$

(III) Stop when step (II) can no longer be implemented

Fleury's Algorithm (1883)

(I) $v_0 \in V(G)$; $\quad P_0 = v_0$

(II) Suppose $C_i = v_0 e_1 v_1 e_2 \cdots e_i v_i$.

Choose e_{i+1} from $E(G) - \{e_1, e_2, \cdots, e_i\}$ as follows:

(a) e_{i+1} is incident with v_i

(b) Unless there is no alternative, e_{i+1} is not a bridge of $G - \{e_1, e_2, \cdots, e_i\}$

(III) Stop when step (II) can no longer be implemented

Fleury's Algorithm (1883)

(I) $v_0 \in V(G)$; $P_0 = v_0$

(II) Suppose $C_i = v_0 e_1 v_1 e_2 \cdots e_i v_i$.

Choose e_{i+1} from $E(G) - \{e_1, e_2, \cdots, e_i\}$ as follows:

(a) e_{i+1} is incident with v_i

(b) Unless there is no alternative, e_{i+1} is not a bridge of $G - \{e_1, e_2, \cdots, e_i\}$

(III) Stop when step (II) can no longer be implemented

Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i *is incident with* ≤ 1 *bridge in* $E(G) - \{e_1, e_2, \cdots, e_i\}$.

Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i is incident with ≤ 1 bridge in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

By Contradiction.

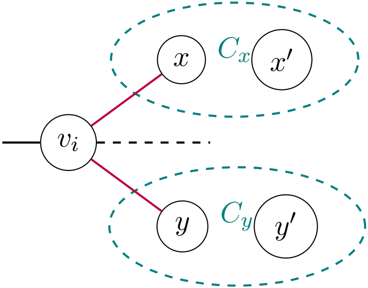
Suppose that v_i is incident with ≥ 2 bridges in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i is incident with ≤ 1 bridge in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

By Contradiction.

Suppose that v_i is incident with ≥ 2 bridges in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

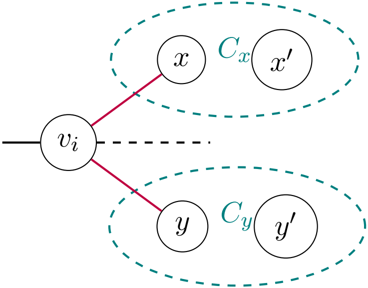


Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i is incident with ≤ 1 bridge in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

By Contradiction.

Suppose that v_i is incident with ≥ 2 bridges in $E(G) - \{e_1, e_2, \cdots, e_i\}$.



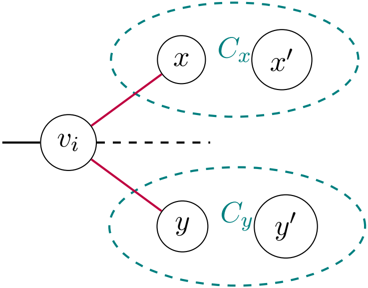
$\exists x' \in C_x : \deg(x)$ is odd

Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i is incident with ≤ 1 bridge in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

By Contradiction.

Suppose that v_i is incident with ≥ 2 bridges in $E(G) - \{e_1, e_2, \cdots, e_i\}$.



$\exists x' \in C_x : \deg(x)$ is odd

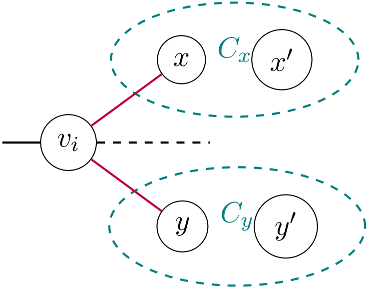
$\exists y' \in C_y : \deg(y)$ is odd

Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i is incident with ≤ 1 bridge in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

By Contradiction.

Suppose that v_i is incident with ≥ 2 bridges in $E(G) - \{e_1, e_2, \cdots, e_i\}$.



$\exists x' \in C_x : \deg(x)$ is odd

$\exists y' \in C_y : \deg(y)$ is odd

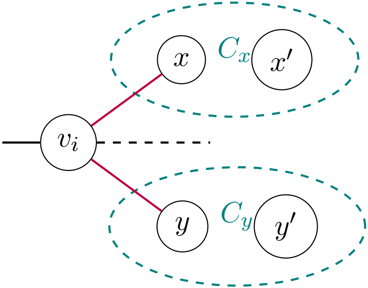
We have found 2 odd vertices.

Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i is incident with ≤ 1 bridge in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

By Contradiction.

Suppose that v_i is incident with ≥ 2 bridges in $E(G) - \{e_1, e_2, \cdots, e_i\}$.



$$\exists x' \in C_x : \deg(x) \text{ is odd}$$

$$\exists y' \in C_y : \deg(y) \text{ is odd}$$

We have found 2 odd vertices.

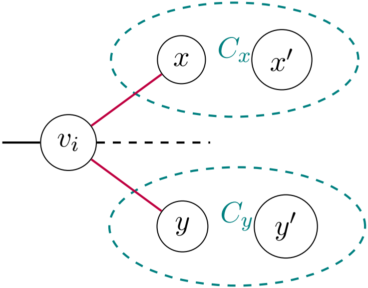
Q : What is the contradiction?

Theorem (Bridges in Fleury's Algorithm)

At any stage, v_i is incident with ≤ 1 bridge in $E(G) - \{e_1, e_2, \cdots, e_i\}$.

By Contradiction.

Suppose that v_i is incident with ≥ 2 bridges in $E(G) - \{e_1, e_2, \cdots, e_i\}$.



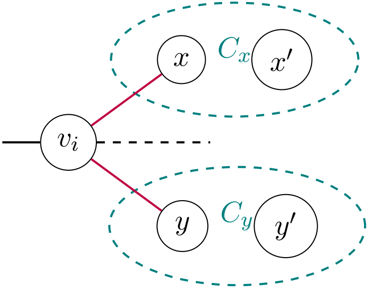
$\exists x' \in C_x : \deg(x)$ is odd

$\exists y' \in C_y : \deg(y)$ is odd

We have found 2 odd vertices.

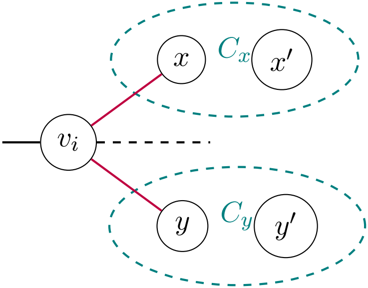
Q : What is the contradiction?

Is $\deg(v_i)$ odd or even?



We have found 2 odd vertices.

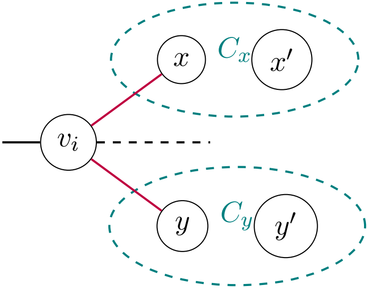
Is $\deg(v_i)$ odd or even?



Case I: $\deg(v_i)$ is odd.

We have found 2 odd vertices.

Is $\deg(v_i)$ odd or even?

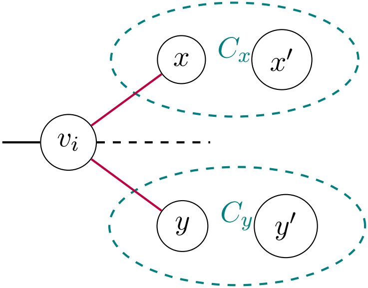


We have found 2 odd vertices.

Is $\deg(v_i)$ odd or even?

CASE I: $\deg(v_i)$ is odd.

Contradiction:



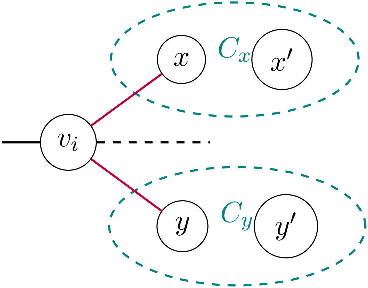
We have found 2 odd vertices.

Is $\deg(v_i)$ odd or even?

CASE I: $\deg(v_i)$ is odd.

Contradiction:

Only v_0 and v_i can have odd degrees!



We have found 2 odd vertices.

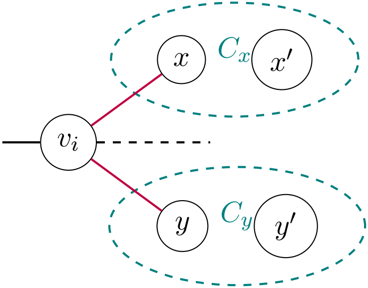
Is $\deg(v_i)$ odd or even?

CASE I: $\deg(v_i)$ is odd.

Contradiction:

Only v_0 and v_i can have odd degrees!

CASE II: $\deg(v_i)$ is even.



We have found 2 odd vertices.

Is $\deg(v_i)$ odd or even?

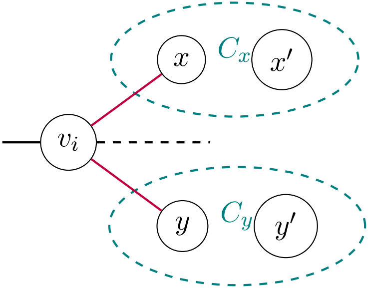
CASE I: $\deg(v_i)$ is odd.

Contradiction:

Only v_0 and v_i can have odd degrees!

CASE II: $\deg(v_i)$ is even.

$$v_i = v_0$$



We have found 2 odd vertices.

Is $\deg(v_i)$ odd or even?

Case I: $\deg(v_i)$ is odd.

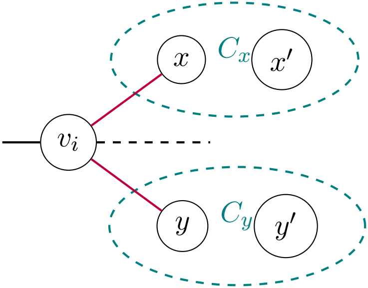
Contradiction:

Only v_0 and v_i can have odd degrees!

Case II: $\deg(v_i)$ is even.

$$v_i = v_0$$

Contradiction:



We have found 2 odd vertices.

Is $\deg(v_i)$ odd or even?

CASE I: $\deg(v_i)$ is odd.

Contradiction:

Only v_0 and v_i can have odd degrees!

CASE II: $\deg(v_i)$ is even.

$$v_i = v_0$$

Contradiction: No odd vertices!

procedure FLEURY(G)

$v_0 \in V(G)$ ▷ Choose any starting vertex

$C \leftarrow v_0$ ▷ Keep track of the circuit

$i \leftarrow 0,\ V_0 \leftarrow V(G),\ E_0 \leftarrow E(G)$

procedure FLEURY(G)
    $v_0 \in V(G)$ ▷ Choose any starting vertex
    $C \leftarrow v_0$ ▷ Keep track of the circuit
    $i \leftarrow 0,\ V_0 \leftarrow V(G),\ E_0 \leftarrow E(G)$

    while $\deg(v_i) > 0$ in E_i **do** ▷ Stop otherwise

    return C

procedure FLEURY(G)

  $v_0 \in V(G)$ ▷ Choose any starting vertex

  $C \leftarrow v_0$ ▷ Keep track of the circuit

  $i \leftarrow 0,\ V_0 \leftarrow V(G),\ E_0 \leftarrow E(G)$

  while $\deg(v_i) > 0$ in E_i **do** ▷ Stop otherwise

    if $\deg(v_i) = 1$ in E_i **then** ▷ No alternative: go the bridge

      $e_{i+1} \triangleq v_i v_{i+1}$

      ▷ Delete the isolated vertex v_i

    else ▷ Have alternatives: don't go the bridge

      Choose $e_{i+1} \triangleq v_i v_{i+1}$ that is not a bridge of (V_i, E_i)

      ▷ No isolated vertex produced

  return C

procedure FLEURY(G)
  $v_0 \in V(G)$ ▷ Choose any starting vertex
  $C \leftarrow v_0$ ▷ Keep track of the circuit
  $i \leftarrow 0,\ V_0 \leftarrow V(G),\ E_0 \leftarrow E(G)$

  while $\deg(v_i) > 0$ in E_i **do** ▷ Stop otherwise
    if $\deg(v_i) = 1$ in E_i **then** ▷ No alternative: go the bridge
      $e_{i+1} \triangleq v_i v_{i+1}$
      ▷ Delete the isolated vertex v_i
    else ▷ Have alternatives: don't go the bridge
      Choose $e_{i+1} \triangleq v_i v_{i+1}$ that is not a bridge of (V_i, E_i)
      ▷ No isolated vertex produced

    $C \leftarrow C e_{i+1} v_{i+1}$
    $E_{i+1} \leftarrow E_i - \{e_{i+1}\}$
    $i \leftarrow i + 1$

  return C

procedure FLEURY(G)

  $v_0 \in V(G)$ ▷ Choose any starting vertex

  $C \leftarrow v_0$ ▷ Keep track of the circuit

  $i \leftarrow 0,\ V_0 \leftarrow V(G),\ E_0 \leftarrow E(G)$

  while $\deg(v_i) > 0$ in E_i **do** ▷ Stop otherwise

    if $\deg(v_i) = 1$ in E_i **then** ▷ No alternative: go the bridge

      $e_{i+1} \triangleq v_i v_{i+1}$

      $V_{i+1} \leftarrow V_i - \{v_i\}$ ▷ Delete the isolated vertex v_i

    else ▷ Have alternatives: don't go the bridge

      Choose $e_{i+1} \triangleq v_i v_{i+1}$ that is not a bridge of (V_i, E_i)

      $V_{i+1} \leftarrow V_i$ ▷ No isolated vertex produced

    $C \leftarrow C e_{i+1} v_{i+1}$

    $E_{i+1} \leftarrow E_i - \{e_{i+1}\}$

    $i \leftarrow i + 1$

  return C

PROOF

We need to prove that

PROOF

We need to prove that C returned by FLEURY is an Eulerian circuit.

PROOF

We need to prove that C returned by FLEURY is an Eulerian circuit.

Definition (Eulerian Circuit)

A connected graph is Eulerian if there exists a closed trail that includes every edge of G.

A trail is a walk in which all the edges are distinct.

PROOF

We need to prove that C returned by FLEURY is an Eulerian circuit.

Definition (Eulerian Circuit)

A connected graph is Eulerian if there exists a closed trail that includes every edge of G.

A trail is a walk in which all the edges are distinct.

Closed Trail Include every edge of G

PROOF

We need to prove that C returned by FLEURY is an Eulerian circuit.

Definition (Eulerian Circuit)

A connected graph is Eulerian if there exists a closed trail that includes every edge of G.

A trail is a walk in which all the edges are distinct.

Closed	Trail	Include every edge of G

$\because$ even degrees

PROOF

We need to prove that C returned by FLEURY is an Eulerian circuit.

Definition (Eulerian Circuit)

A connected graph is Eulerian if there exists a closed trail that includes every edge of G.

A trail is a walk in which all the edges are distinct.

Closed	Trail	Include every edge of G
$\because$ even degrees	$\because$ used edges are deleted	

PROOF

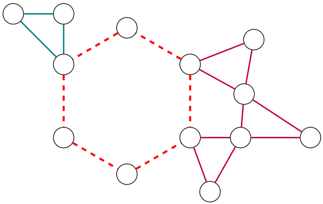
We need to prove that C returned by FLEURY is an Eulerian circuit.

Definition (Eulerian Circuit)

A connected graph is Eulerian if there exists a closed trail that includes every edge of G.

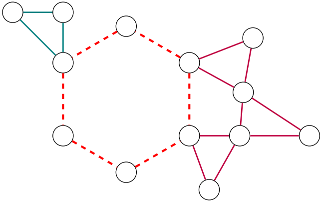
A trail is a walk in which all the edges are distinct.

Closed	Trail	Include every edge of G
$\because$ even degrees	$\because$ used edges are deleted	By Contradiction.



Include every edge of G

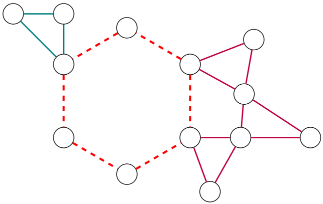
By Contradiction.



Include every edge of G

By Contradiction.

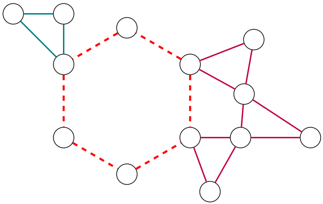
We know that $C : v_0 \sim v_0$



Include every edge of G

By Contradiction.

We know that $C : v_0 \sim v_0$ $\quad E' \triangleq E(G) - E(C) \neq \emptyset$

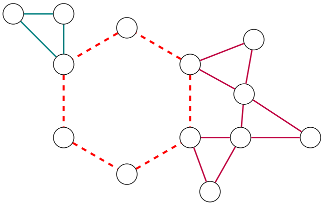


Include every edge of G

By Contradiction.

We know that $C : v_0 \sim v_0$ $\qquad E' \triangleq E(G) - E(C) \neq \emptyset$

$\deg(v_0) = 0$

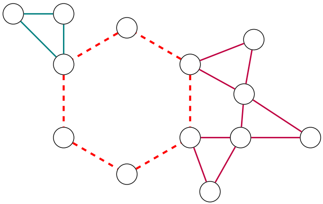


Include every edge of G

By Contradiction.

We know that $C : v_0 \sim v_0$ $\qquad E' \triangleq E(G) - E(C) \neq \emptyset$

$\deg(v_0) = 0$ (Otherwise, Fleury is not terminated.)



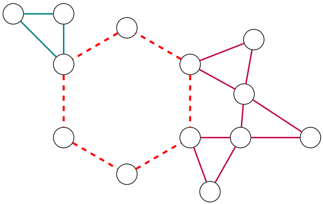
Include every edge of G

By Contradiction.

We know that $C : v_0 \sim v_0$ $\qquad E' \triangleq E(G) - E(C) \neq \emptyset$

$\deg(v_0) = 0$ (Otherwise, FLEURY is not terminated.)

$G|_{E'}$ is disconnected from v_0



Include every edge of G

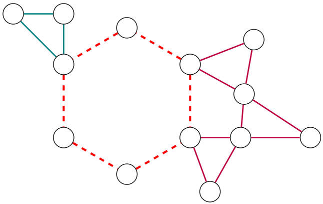
By Contradiction.

We know that $C : v_0 \sim v_0$ $\qquad E' \triangleq E(G) - E(C) \neq \emptyset$

$\deg(v_0) = 0$ (Otherwise, FLEURY is not terminated.)

$G|_{E'}$ is disconnected from v_0

Impossible:



Include every edge of G

By Contradiction.

We know that $C : v_0 \sim v_0$ $\quad$ $E' \triangleq E(G) - E(C) \neq \emptyset$

$\deg(v_0) = 0$ (Otherwise, FLEURY is not terminated.)

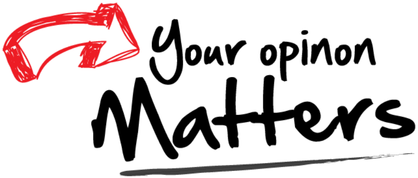
$G|_{E'}$ is disconnected from v_0

Impossible:

(I) Don't go the bridge unless there is no alternative.

(II) Delete the isolated vertex left by going the bridge.


THANK YOU



Office 302

Mailbox: H016

hfwei@nju.edu.cn