VNU-HUS MAT3500: Toán rời rạc

Lý thuyết đồ thị II

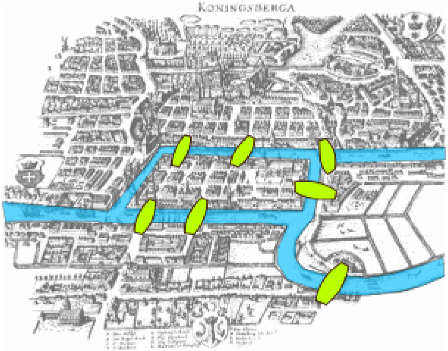
Đường đi ngắn nhất, Đồ thị phẳng, Tô màu đồ thị

Hoàng Anh Đức

Bộ môn Tin học, Khoa Toán-Cơ-Tin học
Đại học KHTN, ĐHQG Hà Nội
hoanganhduc@hus.edu.vn

Nội dung

Hình: Leonhard Euler 1707–1783 (Wikipedia)



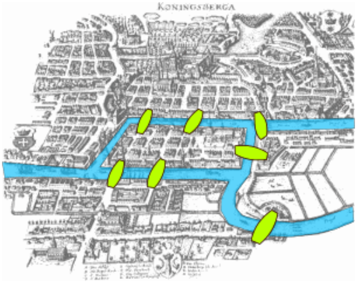
Hình: Bản đồ Königsberg cũ (Wikipedia)

Bảy cây cầu ở Königsberg

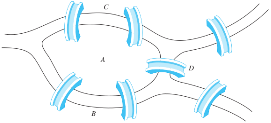
Tìm một tuyến đường đi qua mỗi cây cầu chính xác một lần và quay lại vị trí xuất phát

Đường đi Euler và Đường đi Hamilton

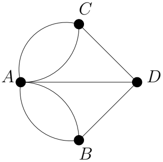
Bảy cây cầu ở Königsberg



(a) Bản đồ Königsberg cũ (Wikipedia)



(b) Bản đồ Königsberg cũ đơn giản hóa

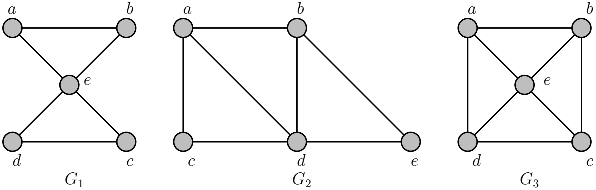


(c) Đồ thị tương ứng

- Đồ thị tương ứng:
  - Mỗi vùng đất ứng với một đỉnh
  - Mỗi cây cầu nối hai vùng đất ứng với một cạnh
- Tìm chu trình đơn trong đồ thị chứa tất cả các cạnh

Cho đồ thị $G = (V, E)$. Một *đường đi/chu trình Euler (Eulerian path/circuit)* trong G là một đường đi/chu trình đơn có chứa mọi cạnh của G

Ví dụ 1



- G_1 có chu trình Euler, G_2 và G_3 không có
- G_2 có đường đi Euler, G_3 không có

Bài tập 1

Chứng minh rằng nếu $G = (V, E)$ là một đa đồ thị vô hướng thỏa mãn $\deg_G(u) \geq 2$ với mọi $u \in V$ thì G có một chu trình đơn

Định lý 1: [Euler 1736]; [Hierholzer and Wiener 1873]

Một đa đồ thị vô hướng liên thông có một chu trình Euler khi và chỉ khi mỗi đỉnh của đồ thị có bậc chẵn

Chứng minh.

($\Rightarrow$) Giả sử một đa đồ thị vô hướng liên thông $G = (V, E)$ có một chu trình Euler $e_1, e_2, \dots, e_m$ trong đó $e_i = x_{i-1}x_i \in E$ với $1 \leq i \leq m$ và $x_0 = x_m = u$.

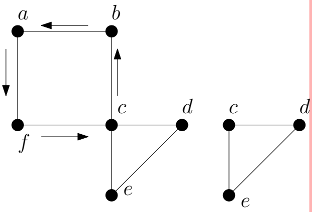
- Với $v = x_i$ $(2 \leq i \leq m-1)$: chu trình đi vào v qua e_i và đi ra qua e_{i+1}
- Với $u = x_0 = x_m$: chu trình đi ra u qua e_1 và trở lại u qua e_m

$\square$

Chứng minh (tiếp).

($\Leftarrow$) Giả sử mọi đỉnh của G đều có bậc chẵn. Lặp lại quá trình chọn chu trình sau cho đến khi đã chọn hết các cạnh (**Thuật toán Hierholzer (1873)**)

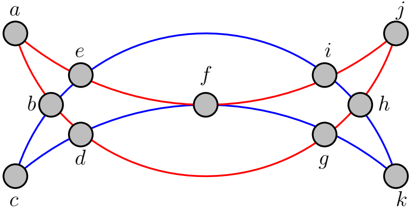
- Xuất phát từ đỉnh $x_0 = a$ bất kỳ
- Xây dựng một đường đi đơn bằng cách chọn tùy ý các cạnh $x_0x_1, x_1x_2, \dots, x_{k-1}x_k$ để thêm vào đường đi cho đến khi không chọn được nữa
- Do bậc của mỗi đỉnh là chẵn, với mỗi đỉnh x_i, ta luôn có thể đi vào từ cạnh $x_{i-1}x_i$ và đi ra từ cạnh x_ix_{i+1}. Do bậc của a cũng phải là chẵn, cạnh cuối cùng được chọn sẽ có dạng ya
- Bỏ đi các cạnh đã chọn và các đỉnh không kề với các cạnh còn lại



Cuối cùng, ghép các chu trình trên thành một chu trình Euler. (Thuật toán chạy trong thời gian $O(|E|)$)

Ví dụ 2

Tìm chu trình Euler trong đồ thị sau



- Bắt đầu từ $x_0 = a$, chọn tùy ý các cạnh $x_0x_1, x_1x_2, \dots,$ $x_{k-1}x_k, x_ka$. Ví dụ: $ae, ef, fi, ij, jh, hg, gd, db, ba$
- Bỏ đi các cạnh đã chọn và các đỉnh a, j
- Bắt đầu từ $x_0 = c$, chọn tùy ý các cạnh $x_0x_1, x_1x_2, \dots,$ $x_{l-1}x_l, x_lc$. Ví dụ: $cb, be, ei, ih, hk, kg, gf, fd, dc$
- Bỏ đi các cạnh đã chọn và các đỉnh cô lập còn lại
- Ghép hai chu trình đã chọn thành một chu trình Euler:
  - $ae, ei, ih, hk, kg, gf, fd, dc, cb, be, ef, fi, ij, jh, hg, gd, db, ba$
  - $ae, ef, fd, dc, cb, be, ei, ih, hk, kg, gf, fi, ij, jh, hg, gd, db, ba$
  - $\dots$

Bài tập 2

Với những giá trị nào của n thì các đồ thị sau có chu trình Euler?

(a) K_n

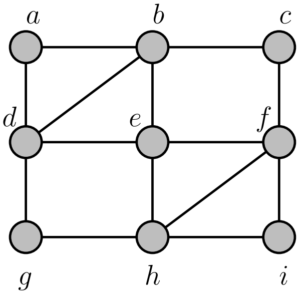
(b) C_n

(c) W_n

(d) Q_n

Bài tập 3

Tìm chu trình Euler trong đồ thị sau



Định lý 2

Một đa đồ thị vô hướng liên thông G có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi có đúng hai đỉnh của G có bậc lẻ.

Chứng minh.

($\Rightarrow$) Giả sử G có đường đi Euler nhưng không có chu trình Euler

- Hai đỉnh ở hai đầu mút của đường đi có bậc lẻ
- Các đỉnh còn lại có bậc chẵn

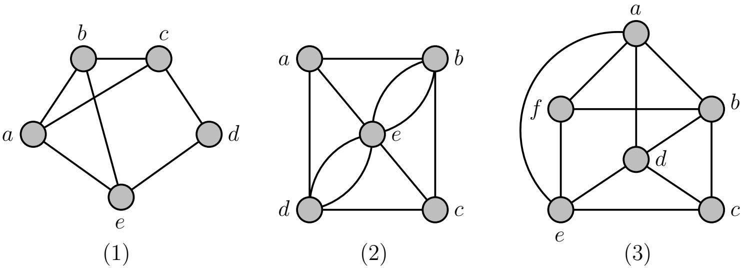
($\Leftarrow$) Giả sử G có chính xác hai đỉnh bậc lẻ u, v

- Tìm chu trình Euler của đồ thị $G + uv$
- Xóa cạnh uv trong chu trình để thu được đường đi Euler trong G

$\square$

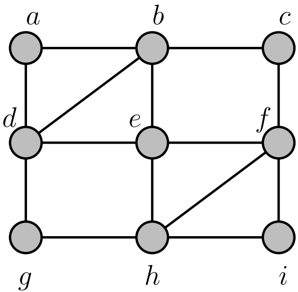
Bài tập 4

Hãy xác định xem các đồ thị sau có chu trình/đường đi Euler hay không? Nếu có, hãy tìm một chu trình/đường đi Euler trong đồ thị đó



Bài tập 5

Thuật toán Fleury [Fleury 1883] xây dựng các chu trình Euler bằng cách đầu tiên chọn một đỉnh tùy ý của một đa đồ thị liên thông, và sau đó tạo thành một chu trình bằng cách chọn các cạnh liên tiếp. Một khi một cạnh được chọn, nó sẽ bị loại bỏ. Các cạnh được chọn liên tiếp sao cho mỗi cạnh bắt đầu từ nơi cạnh cuối cùng kết thúc, và sao cho cạnh này không phải là một cạnh cắt trừ khi không có lựa chọn thay thế nào khác. Sử dụng thuật toán Fleury để tìm một chu trình Euler trong đồ thị sau



Tương tự, đối với đồ thị có hướng $G = (V, E)$, một *đường đi/chu trình Euler (Eulerian path/circuit)* trong G là một đường đi/chu trình đơn có hướng chứa mọi cạnh của G

Định lý 3

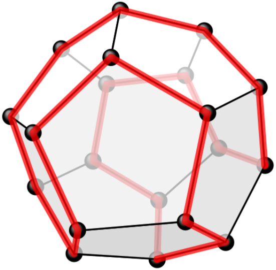
Một đa đồ thị có hướng liên thông mạnh có chu trình Euler khi và chỉ khi mọi đỉnh của đồ thị có bậc vào bằng bậc ra

- Điều kiện "liên thông mạnh" có thể được thay bằng "liên thông yếu" (*Tại sao?*)

Định lý 4

Một đa đồ thị có hướng liên thông yếu có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi có đúng hai đỉnh u, v của G sao cho $\deg^-(u) = \deg^+(u) + 1$ và $\deg^-(v) = \deg^+(v) - 1$

Hình: Sir William Rowan Hamilton 1805-1865 (Wikipedia)



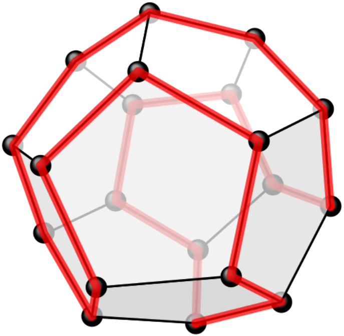
Hình: Trò chơi “Vòng quanh thế giới” (Wikipedia)

Trò chơi “Vòng quanh thế giới”

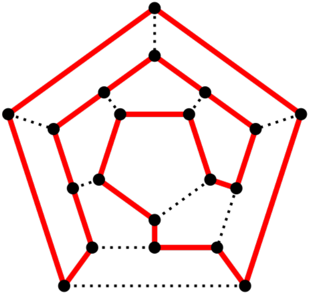
Mỗi đỉnh trong 20 đỉnh của khối 12 mặt đại diện cho một thành phố. Tìm đường đi xuất phát từ một đỉnh dọc theo các cạnh của khối, ghé thăm mỗi đỉnh còn lại một lần, và quay lại vị trí ban đầu

Đường đi Euler và Đường đi Hamilton

Trò chơi "Vòng quanh thế giới" (1857)



(a) Trò chơi "Vòng quanh thế giới" (Wikipedia)

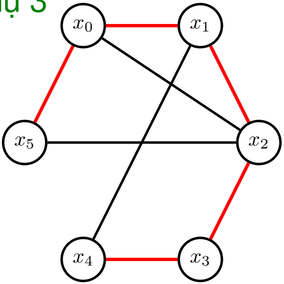


(b) Đồ thị đẳng cấu với khối 12 mặt (Wikipedia)

Cho $G = (V, E)$ là một đồ thị vô hướng gồm $n \geq 1$ đỉnh

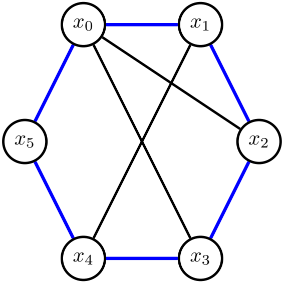
- Một *đường đi Hamilton* trong G là một đường đi đơn $x_0, x_1, \dots, x_{n-1}$ thỏa mãn điều kiện $V = \{x_0, x_1, \dots, x_{n-1}\}$ và $x_i \neq x_j$ với $0 \leq i < j \leq n-1$
- Một *chu trình Hamilton* trong G là một chu trình đơn x_0, x_1, x_{n-1}, x_0 thỏa mãn điều kiện x_0, x_1, x_{n-1} là một đường đi Hamilton

Ví dụ 3



Đường đi Hamilton

$x_5, x_0, x_1, x_2, x_3, x_4$

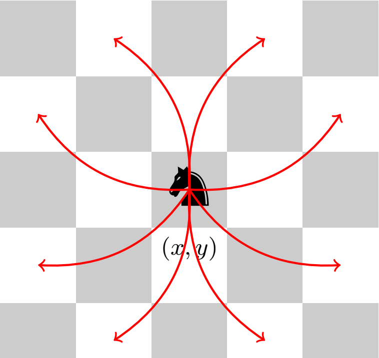


Chu trình Hamilton

$x_0, x_1, x_2, x_3, x_4, x_5, x_0$

Ví dụ 4 (Bài toán quân mã đi tuần (Knight's tour))

Quân mã (knight) là một quân cờ có thể di chuyển hai ô theo chiều ngang và một ô theo chiều dọc, hoặc một ô theo chiều ngang và hai ô theo chiều dọc trên bàn cờ. Nghĩa là, một quân mã tại ô (x, y) có thể di chuyển đến một trong tám ô $(x \pm 2, y \pm 1), (x \pm 1, y \pm 2)$, nếu những ô này nằm trên bàn cờ, như được minh họa dưới đây

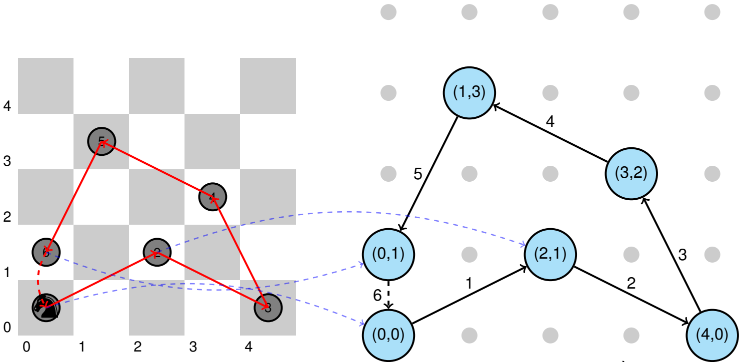


Ví dụ 4 (tiếp)

- *Đường đi tuần của quân mã (knight's tour)* là một chuỗi các nước đi hợp lệ của quân mã bắt đầu từ một ô nào đó và thăm mỗi ô đúng một lần ⇔ *đường đi Hamilton* trong đồ thị tương ứng
- Đường đi tuần được gọi là *khép kín (reentrant)* nếu có một nước đi hợp lệ đưa quân mã từ ô cuối cùng trở lại ô xuất phát ⇔ *chu trình Hamilton* trong đồ thị tương ứng
- Có thể mô hình hóa bằng cách sử dụng đồ thị:
  - Một đỉnh cho mỗi ô trên bàn cờ
  - Một cạnh nối hai đỉnh nếu quân mã có thể di chuyển hợp lệ giữa hai ô tương ứng với các đỉnh đó

Đường đi Euler và Đường đi Hamilton

Đường đi Hamilton



(a) Bàn cờ 5×5 với một vài bước nhảy của quân mã

(b) Mô hình đồ thị tương ứng

Hình: Một số bước đi của quân mã trên bàn cờ 5×5 và các cạnh tương ứng trên đồ thị được mô tả. Quân mã xuất phát từ góc dưới cùng bên trái (ô $(0,0)$). Mũi tên với nét liền (nét đứt) thể hiện quân mã nhảy (không nhảy được) được từ ô ban đầu đến ô kế tiếp

Bài tập 6

(a) Vẽ đồ thị biểu diễn các nước đi hợp lệ của quân mã trên bàn cờ 3×3

(b) Vẽ đồ thị biểu diễn các nước đi hợp lệ của quân mã trên bàn cờ 3×4

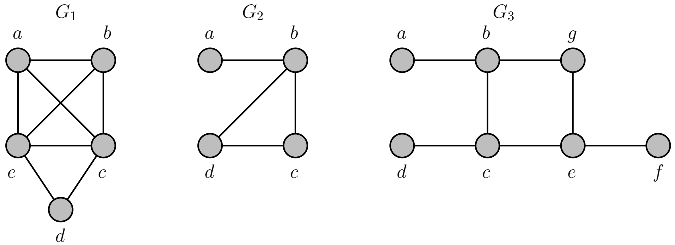
Bài tập 7

Chứng minh rằng đồ thị biểu diễn các nước đi hợp lệ của quân mã trên bàn cờ kích thước $m \times n$, với mọi số nguyên dương m và n, là đồ thị hai phần

- *Chưa có* một thuật toán trong thời gian đa thức để kiểm tra xem một đồ thị có chu trình Hamilton hay không
- Một số tính chất có thể được sử dụng để chỉ ra một đồ thị *không có chu trình Hamilton*
  - Đồ thị *có chứa đỉnh bậc* 1 không có chu trình Hamilton
  - Nếu *đỉnh* v của đồ thị G có *bậc* 2 thì *hai cạnh kề với* v *thuộc mọi chu trình Hamilton của* G *(nếu có)*
  - Một chu trình Hamilton *không chứa một chu trình con nào có số đỉnh nhỏ hơn nó*

Bài tập 8

Các đồ thị sau có chu trình/đường đi Hamilton không?



Bài tập 9

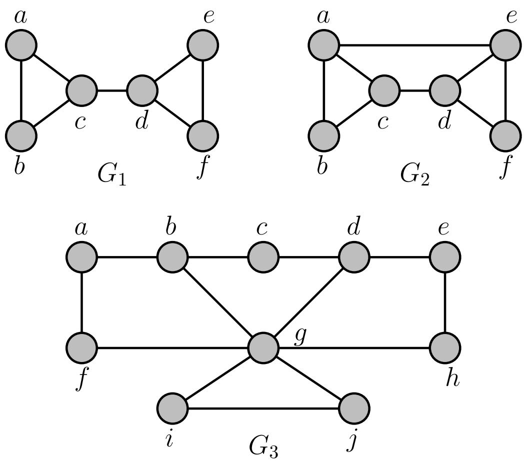
Hãy cho ví dụ về một đồ thị mà chu trình Euler của nó cũng là chu trình Hamilton

Bài tập 10

Một thông báo chẩn đoán (diagnostic message) có thể được gửi qua mạng máy tính để thực hiện kiểm tra trên tất cả các liên kết và trong tất cả các thiết bị. Loại đường đi nào nên được sử dụng để kiểm tra tất cả các liên kết? Để kiểm tra tất cả các thiết bị?

Bài tập 11

Đồ thị nào sau đây có chu trình Hamilton? Tại sao?



Định lý 5: Định lý Dirac

Nếu $G = (V, E)$ là một đơn đồ thị vô hướng gồm n đỉnh ($n \geq 3$) thỏa mãn điều kiện bậc của mỗi đỉnh trong G lớn hơn hoặc bằng $n/2$ thì G có một chu trình Hamilton

Bài tập 12 (($\star$) Chứng minh Định lý Dirac)

(a) Dễ thấy Định lý đúng với $n = 3$. Giả sử $n \geq 4$

(b) G phải liên thông (Tại sao?)

(c) Gọi $P = v_0, v_1, \dots, v_k$ là đường đi đơn có độ dài lớn nhất trong G ($0 \leq k \leq n-1$).
- Mọi đỉnh kề với v_0 hoặc v_k đều phải thuộc P (Tại sao?)
- Do $\deg(v_k) \geq n/2$, có ít nhất $n/2$ cạnh phân biệt v_iv_{i+1} của P thỏa mãn $v_iv_k \in E$. Tương tự, do $\deg(v_0) \geq n/2$, có ít nhất $n/2$ cạnh phân biệt v_jv_{j+1} của P thỏa mãn $v_0v_{j+1} \in E$
- Do P có ít hơn n cạnh, tồn tại một cạnh v_qv_{q+1} thỏa mãn đồng thời hai điều kiện trên: $v_qv_k \in E$ và $v_0v_{q+1} \in E$

(d) P chứa tất cả các đỉnh của G (Tại sao?)

Định lý 6: Định lý Ore

Nếu $G = (V, E)$ là một đơn đồ thị vô hướng gồm n đỉnh ($n \geq 3$) thỏa mãn điều kiện $\deg(u) + \deg(v) \geq n$ với mọi cặp đỉnh u, v không kề nhau trong G thì G có một chu trình Hamilton

Bài tập 13

Chứng minh Định lý Dirac (Định lý 5) bằng cách sử dụng Định lý Ore

Bài tập 14 ($\star$)

Cho $G = (V_1 \cup V_2, E)$ là một đồ thị hai phần với $|V_1| = |V_2| = n$ ($n \geq 2$). Chứng minh rằng nếu $\deg(v) > n/2$ với mọi đỉnh $v \in V = V_1 \cup V_2$ thì G có một chu trình Hamilton (**Gợi ý:** Xem lại chứng minh của Định lý Dirac.)

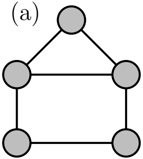
Bài tập 15

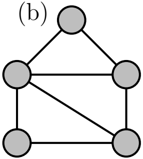
Với mỗi đồ thị sau, hãy xác định

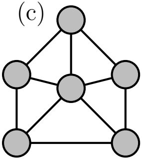
(i) có thể sử dụng Định lý Dirac để chứng minh đồ thị có chu trình Hamilton không?

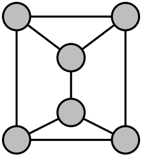
(ii) có thể sử dụng Định lý Ore để chứng minh đồ thị có chu trình Hamilton không?

(iii) đồ thị có chu trình Hamilton không?

(a)


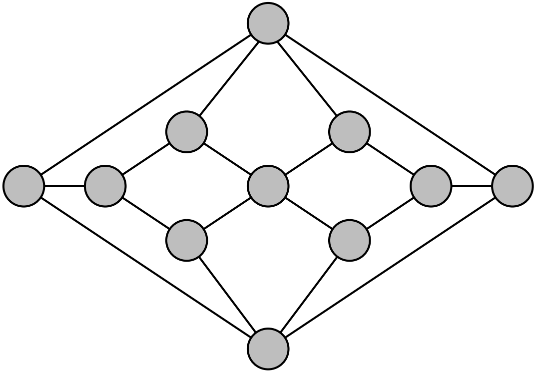
(b)


(c)


(d)


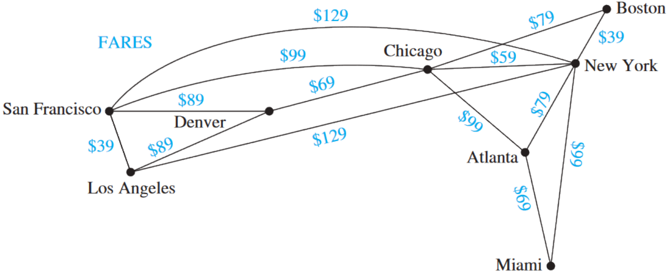
Bài tập 16

Chứng minh rằng đồ thị sau đây không có chu trình Hamilton. (**Gợi ý:** Đồ thị đã cho là một đồ thị hai phần)



Bài toán đường đi ngắn nhất
Đồ thị có trọng số

- Một *đồ thị có trọng số (weighted graph)* $G = (V, E, w)$ gồm tập đỉnh V, tập cạnh E, và một hàm $w : E \to \mathbb{R}$ gán mỗi cạnh (cung) $e \in E$ bởi một số thực $w(e)$ gọi là *trọng số (weight)* của cạnh (cung) e



Hình: Đồ thị có trọng số mô tả giá vé của các chuyến bay giữa một số thành phố ở Mỹ (từ [Rosen 2012], Chương 10)

Bài toán đường đi ngắn nhất

Đồ thị có trọng số

Cho $G = (V, E, w)$ là đơn đồ thị có trọng số

- Một đường đi từ u đến v qua các cạnh (cung) $e_1, e_2, \dots, e_n$ có *độ dài (length)* $c(u, v) = \sum_{i=1}^{n} w(e_i)$
- *Khoảng cách (distance)* giữa hai đỉnh u, v, ký hiệu $d_G(u, v)$, là chiều dài nhỏ nhất của một đường đi từ u đến v

Bài toán đường đi ngắn nhất

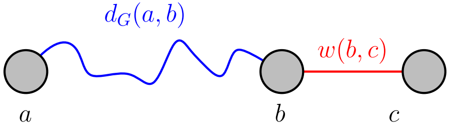
- **Input:** Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó $V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \to \mathbb{R} \cup \{\infty\}$ với $w(v_i, v_j) = \infty$ nếu $v_iv_j \notin E$
- **Output:** Khoảng cách $d_G(a, z)$

Bài toán đường đi ngắn nhất

- **Input:** Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó $V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \to \mathbb{R} \cup \{\infty\}$ với $w(v_i, v_j) = \infty$ nếu $v_iv_j \notin E$
- **Output:** Khoảng cách $d_G(a, z)$

Thuật toán Dijkstra:

- **Điều kiện:** *Trọng số dương*, nghĩa là $w(v_i, v_j) > 0$ với mọi $0 \leq i < j \leq n$ (hay $w : [V]^2 \to \mathbb{R}^+ \cup \{\infty\}$)
- **Ý tưởng:** Tìm đường đi ngắn nhất từ a tới các đỉnh kế tiếp cho đến khi đạt đến z. Chú ý rằng với các đỉnh a, b, c, *độ dài đường đi ngắn nhất từ a đến c **đi qua đỉnh** b kề với c* bằng *khoảng cách giữa a và b* cộng với *trọng số cạnh nối b và c*



Bài toán đường đi ngắn nhất
Thuật toán Dijkstra

"Làm thế nào để tìm được một con đường ngắn nhất để đi từ Rotterdam đến Groningen, hay nói chung: từ một thành phố đến một thành phố khác? Bằng cách sử dụng thuật toán tìm đường đi ngắn nhất, một thuật toán mà tôi đã thiết kế trong khoảng hai mươi phút. Một buổi sáng, tôi đang mua sắm ở Amsterdam cùng với vợ sắp cưới, và khi mệt mỏi, chúng tôi ngồi xuống một quán cà phê để uống một tách cà phê, và tôi chỉ đang nghĩ về việc liệu tôi có thể trả lời được câu hỏi này không, và sau đó tôi đã thiết kế thuật toán tìm đường đi ngắn nhất. Như tôi đã nói, đó là một phát minh chỉ trong có hai mươi phút. Nó được xuất bản vào năm 1959, ba năm sau đó. Bài báo vẫn còn đọc được, và trên thực tế, nó khá hay. Một trong những lý do khiến nó hay đến vậy là vì tôi đã thiết kế nó mà không cần bút và giấy. Sau này tôi mới biết rằng một trong những lợi ích của việc thiết kế mà không cần bút và giấy là bạn gần như buộc phải tránh tất cả những sự phức tạp không cần thiết."

— Edsger W. Dijkstra, trong một cuộc phỏng vấn với Philip L. Frana, Communications of the ACM, 2001

Thuật toán 1: Thuật toán Dijkstra [Dijkstra 1959]

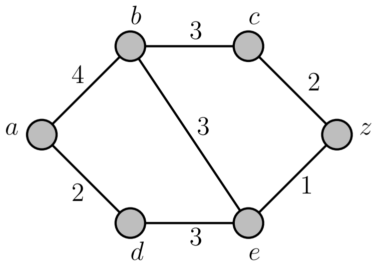
Input: Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó
$V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \to \mathbb{R}^+ \cup \{\infty\}$
với $w(v_i, v_j) = \infty$ nếu $v_i v_j \notin E$

Output: Khoảng cách $d_G(a, z)$

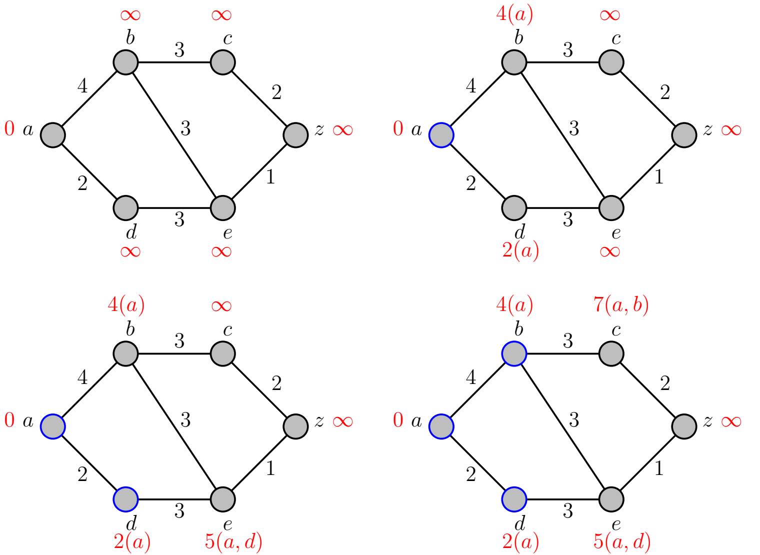
```
Gán nhãn L(a) := 0, L(v_i) := ∞ với mọi v_i ≠ a
S := ∅
while z ∉ S do
    u := đỉnh không thuộc S với L(u) nhỏ nhất
    S := S ∪ {u}
    for v ∉ S do
        if L(u) + w(u, v) < L(v) then
            L(v) := L(u) + w(u, v)
return L(z)
```

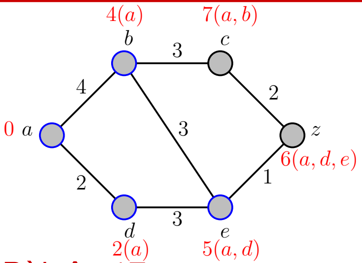
Ví dụ 5

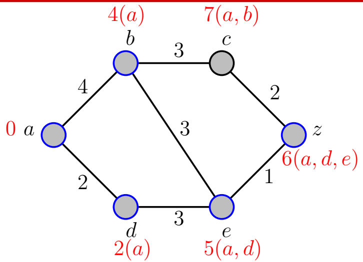
Sử dụng thuật toán Dijkstra để tìm khoảng cách giữa hai đỉnh a và z trong đồ thị sau



Bài toán đường đi ngắn nhất
Thuật toán Dijkstra



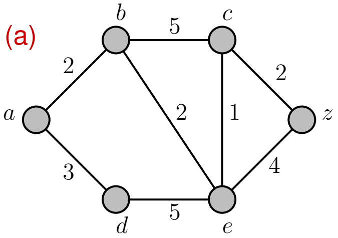




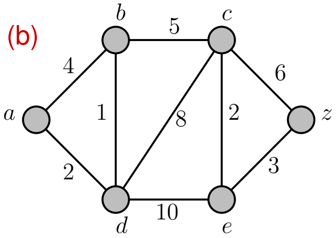
Bài tập 17

Áp dụng thuật toán Dijkstra để tìm khoảng cách từ a đến z trong mỗi đồ thị sau

(a)

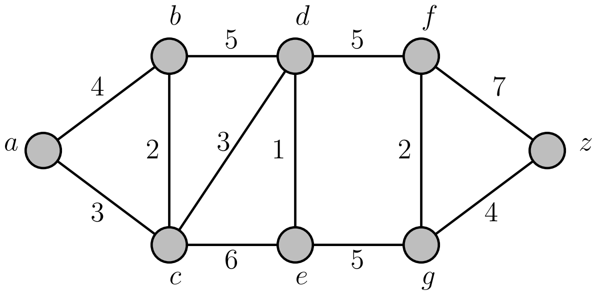


(b)



Bài tập 18

Tìm độ dài đường đi ngắn nhất từ a đến z trong đồ thị sau bằng thuật toán Dijkstra



Bài tập 19

Đường đi ngắn nhất giữa hai đỉnh của một đồ thị có trọng số $G = (V, E, w)$ có phải là duy nhất hay không nếu như trọng số của các cạnh là phân biệt, nghĩa là với hai cạnh $e, f \in E$ bất kỳ thì $w(e) \neq w(f)$?

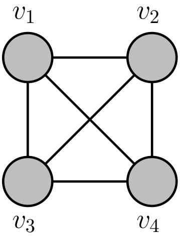
Bài toán đường đi ngắn nhất
Thuật toán Dijkstra

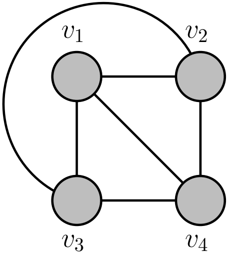
- **Tính đúng đắn:** [Dijkstra 1959]
- **Độ phức tạp:** $O(n^2)$, với $n = |V|$
  - Với cấu trúc dữ liệu "đống Fibonacci" (Fibonacci Heap), thuật toán Dijkstra có thể được lập trình để chạy trong thời gian $O(m + n \log n)$, với $n = |V|$ và $m = |E|$
  - Hiệu quả tốt khi chạy với các "đồ thị thưa" (sparse graph) cực lớn (các đồ thị có m rất nhỏ so với n^2)
- Thuật toán tốt nhất được đề xuất gần đây chạy trong thời gian $O(m \log^{2/3} n)$ [Duan et al. 2025]
- Thuật toán Dijkstra cũng có thể được lập trình để xuất ra một đường đi ngắn nhất từ a đến mỗi đỉnh khác trong đồ thị
- Một cách tiếp cận tìm đường đi khác là thuật toán A* [Hart, Nilsson, and Raphael 1968], thuật toán này mở rộng và cải tiến thuật toán Dijkstra. Thuật toán A* sử dụng heuristic để định hướng việc tìm kiếm, từ đó đạt hiệu năng tốt hơn

- Một đồ thị vô hướng được gọi là *đồ thị phẳng (planar graph)* nếu nó có thể được vẽ trên mặt phẳng sao cho không có hai cạnh nào cắt nhau (ở một điểm không phải là đầu mút của cạnh).
- Hình vẽ như thế được gọi là một *biểu diễn phẳng (planar representation)* của đồ thị.

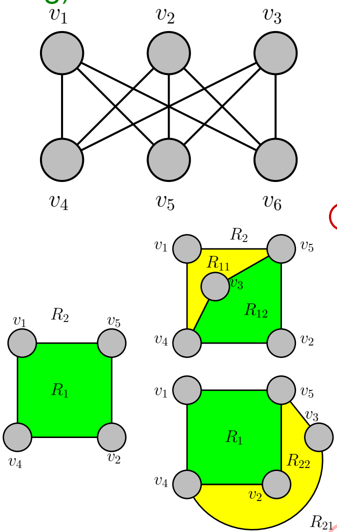
Ví dụ 6

K_4





Ví dụ 7 ($K_{3,3}$ không là đồ thị phẳng)

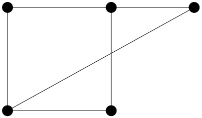


Ta chứng minh bằng phản chứng.
Giả sử $K_{3,3}$ là đồ thị phẳng.

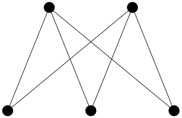
- Trong bất kỳ biểu diễn phẳng nào của $K_{3,3}$ ta có v_1 và v_2 đều phải luôn nối với v_4 và v_5 Các đỉnh này chia mặt phẳng thành hai miền R_1 và R_2
- Đỉnh v_3 thuộc R_1 hoặc R_2
- Vị trí của v_6?

Bài tập 20

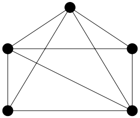
Tìm một biểu diễn phẳng của các đồ thị phẳng sau



G_1



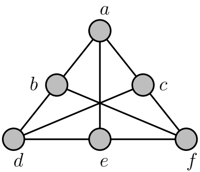
G_2



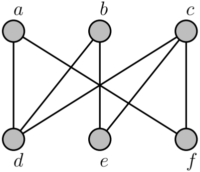
G_3

Bài tập 21

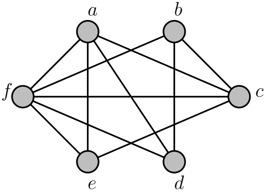
Các đồ thị sau có phải đồ thị phẳng không? Nếu đúng, hãy vẽ một biểu diễn phẳng của đồ thị đó



G_1



G_2



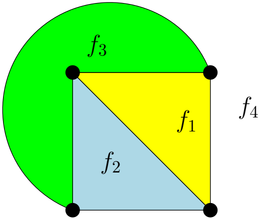
G_3

- Biểu diễn phẳng của một đồ thị phẳng $G = (V, E)$ chia mặt phẳng thành các *miền (region)*, kể cả *miền vô hạn (unbounded region)*
- Hai điểm bất kỳ trong cùng một miền có thể được nối với nhau bằng một nét liền mà không cắt bất kỳ cạnh nào
- *Bậc (degree)* của một miền f, ký hiệu $\deg(f)$, là số cạnh của G trên biên của f

Ví dụ 8

Biểu diễn phẳng của K_4

- chia mặt phẳng thành 4 miền f_1, f_2, f_3, và f_4; và
- $\deg(f_1) = \deg(f_2) = \deg(f_3) = \deg(f_4) = 3$



- **Chú ý:** $\sum_{\text{miền } f \text{ của } G} \deg(f) \leq 2|E|$ (Mỗi cạnh thuộc tối đa hai miền)

Định lý 7: Công thức Euler

Giả sử G là một đơn đồ thị phẳng và liên thông gồm m cạnh, n đỉnh, và r miền. Ta có $n - m + r = 2$

Chứng minh.

- Xây dựng dãy đồ thị $G_1, G_2, \dots, G_m = G$
  - Chọn một cạnh bất kỳ của G làm G_1
  - G_i được tạo thành từ G_{i-1} bằng cách thêm một cạnh bất kỳ liên thuộc với một đỉnh của G_{i-1} $(i \in \{2, 3, \dots, m\})$
  - Gọi n_i, m_i, r_i lần lượt là số đỉnh, cạnh, và miền của một biểu diễn phẳng của G_i
- Công thức Euler đúng với mọi G_i
  - Ta có $n_1 - m_1 + r_1 = 2 - 1 + 1 = 2$
  - Giả sử công thức Euler đúng với G_i, tức là $n_i - m_i + r_i = 2$ Gọi $a_{i+1}b_{i+1}$ là cạnh thêm vào G_i để tạo thành G_{i+1}. Có hai khả năng:
    - một trong hai đỉnh a_{i+1}, b_{i+1} không thuộc G_{i-1}
    - cả a_{i+1} và b_{i+1} thuộc G_{i-1}

Hệ quả 8

Giả sử G là một đồ thị phẳng liên thông gồm m cạnh và n đỉnh ($n \geq 3$). Khi đó, $m \leq 3n - 6$. Thêm vào đó, nếu $m = 3n - 6$ thì mỗi miền của G có chính xác 3 cạnh trên biên.

Chứng minh.

- Nhận xét rằng mỗi miền của G có ít nhất 3 cạnh trên biên, do đó $\sum_{\text{miền } f \text{ của } G} \deg(f) \geq 3r$. Mặt khác, ta cũng có $\sum_{\text{miền } f \text{ của } G} \deg(f) \leq 2m$. Suy ra $3r \leq 2m$.
- Áp dụng công thức Euler, ta có $2 = n - m + r \leq n - m + 2m/3$, suy ra $m \leq 3n - 6$.

$\square$

Bài tập 22

Giả sử G là một đồ thị đơn phẳng và liên thông gồm 20 đỉnh, mỗi đỉnh có bậc 3. Một biểu diễn phẳng của G chia mặt phẳng thành bao nhiêu miền?

Bài tập 23

Trong số các đồ thị không phẳng sau, đồ thị nào thỏa mãn điều kiện sau: nếu bỏ đi một đỉnh bất kỳ và các cạnh liên thuộc với đỉnh đó thì ta thu được một đồ thị phẳng.

(a) K_5
(b) K_6
(c) $K_{3,3}$
(d) $K_{3,4}$

Bài tập 24

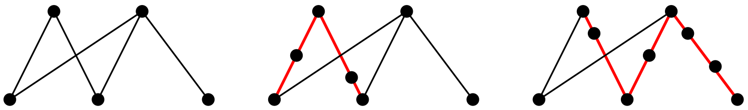
Chứng minh K_5 không là đồ thị phẳng

Bài tập 25

Chứng minh rằng nếu G là một đơn đồ thị phẳng và liên thông thì G có một đỉnh có bậc nhỏ hơn hoặc bằng 5

- Cho đồ thị G. Một *phép phân chia (subdivision)* một cạnh e của G được thực hiện bằng cách thay thế e bằng một đường đi đơn
- Hai đồ thị G_1 và G_2 được gọi là *đồng phôi (homeomorphic)* nếu chúng được xây dựng từ cùng một đồ thị thông qua một dãy các phép phân chia

Ví dụ 9



G $\qquad$ G_1 $\qquad$ G_2

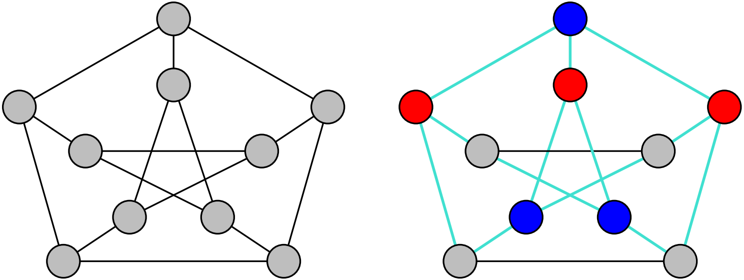
Định lý 9: Định lý Kuratowski

G là đồ thị phẳng khi và chỉ khi nó không chứa bất kỳ đồ thị nào đồng phôi với K_5 hoặc $K_{3,3}$

Có thể hình dung định lý Kuratowski một cách sơ lược như sau:

- Chúng ta biết hai đồ thị không phẳng, đó là $K_{3,3}$ và K_5. Do đó, đương nhiên bất kỳ đồ thị nào chứa những đồ thị này đều không phải là đồ thị phẳng
- Thực tế, bất kỳ đồ thị nào chứa cấu trúc có hình dạng cơ bản giống như hai đồ thị trên đều không phẳng (liên quan đến các khái niệm phép phân chia (subdivision) và đồng phôi (homeomorphic))
- Hơn nữa, mọi đồ thị không phẳng đều chứa một trong hai "hình dạng xấu" này. Đây là cách duy nhất để một đồ thị trở thành không phẳng

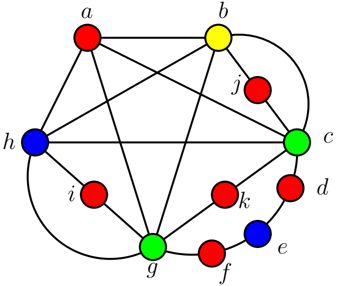
Đồ thị phẳng
Định lý Kuratowski



Hình: Đồ thị Petersen có chứa một đồ thị con đồng phôi với $K_{3,3}$ và do đó không phải là đồ thị phẳng

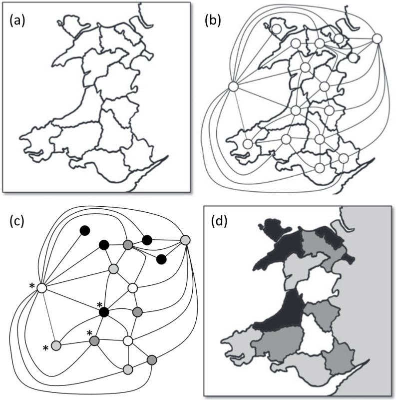
Cho đơn đồ thị vô hướng $G = (V, E)$

- *Tô màu* một đồ thị đơn là sự gán màu cho các đỉnh của đồ thị sao cho không có hai đỉnh liền kề được gán cùng một màu. Cụ thể, với các "màu" $1, 2, \ldots, k$, một *cách tô màu các đỉnh (vertex k-coloring)* của G là một hàm $f : V \to \{1, 2, \ldots, k\}$ thỏa mãn $f(u) \neq f(v)$ với mọi $u, v \in V$ với $uv \in E$



- *Sắc số (chromatic number)* của G, ký hiệu $\chi(G)$, là số tối thiểu các màu cần thiết để tô màu G

Ghi chép sớm nhất về bài toán tô màu đồ thị có lẽ là vào năm 1852 khi Francis Guthrie (1831–1899), lúc đó là một sinh viên ở Đại học Cao đẳng London (University College London), tô màu một bản đồ các quận của Anh và nhận ra là có lẽ chỉ cần bốn màu để tô màu bản đồ sao cho hai quận liền kề nhau có màu khác nhau



- Phỏng đoán của Guthrie được cho là phát biểu đầu tiên của *Định lý bốn màu (Four Color Theorem)*

Định lý 10: Định lý bốn màu

Với mọi đồ thị phẳng G, ta luôn có $\chi(G) \leq 4$

- Năm 1879, Kempe đề xuất một chứng minh cho Định lý bốn màu. Khoảng 10 năm sau đó, Heawood chỉ ra lỗi sai trong chứng minh của Kempe và chỉnh sửa lại chứng minh của Kempe để chỉ ra rằng năm màu là đủ để tô màu bất kỳ đồ thị phẳng nào
- Năm 1976, Kenneth Appel and Wolfgang Haken (Đại học Illinois) [Appel and Haken 1977]; [Appel, Haken, and Koch 1977] chứng minh định lý bốn màu bằng cách giả sử nếu Định lý bốn màu là sai thì sẽ có một phản ví dụ thuộc một trong 1936 loại khác nhau, và chỉ ra rằng không có loại nào dẫn đến phản ví dụ. Các trường hợp này được phân tích cẩn thận nhờ máy tính
- Robertson, Sanders, Seymour, và Thomas [Robertson, Sanders, Seymour, and Thomas 1997] đưa ra một chứng minh đơn giản hơn với 633 loại cần kiểm tra

Tô màu đồ thị

Một số tính chất cơ bản

- Để *chỉ ra $\chi(G) = k$ với đồ thị G nào đó*, ta cần:
  - Chỉ ra một cách tô màu các đỉnh của G bằng k màu (nghĩa là $\chi(G) \leq k$)
  - Chỉ ra rằng không thể dùng ít hơn k màu để tô màu các đỉnh của G (nghĩa là $\chi(G) \geq k$)
- Một số nhận xét
  (1) Mọi đồ thị G gồm n đỉnh có thể được tô màu bằng n màu
  (2) $\chi(K_n) = n$ (Tại sao?)
  (3) Ta ký hiệu $\omega(G)$ là số nguyên dương lớn nhất $r \geq 1$ thỏa mãn K_r là đồ thị con của G. Với mọi đồ thị G, ta có $\omega(G) \leq \chi(G)$. Thông thường, $\omega(G) \neq \chi(G)$
  (4) $\chi(C_n) = 2$ nếu $n \geq 4$ chẵn và $\chi(C_n) = 3$ nếu $n \geq 3$ lẻ (Tại sao?)
  (5) G là đồ thị hai phần khi và chỉ khi $\chi(G) \leq 2$ (Tại sao?)
- *Chưa biết* có tồn tại hay không một thuật toán chạy trong thời gian đa thức để xác định xem một đồ thị G có thể được tô màu bằng 3 màu hay không

Bài tập 26

Tính $\chi(W_n)$, $\chi(K_{m,n})$, và $\chi(Q_n)$

Bài tập 27

Những đồ thị nào có sắc số bằng 1?

Bài tập 28

Một cách tô màu các cạnh của một đồ thị $G = (V, E)$ bằng k màu (*k-edge coloring*) là một hàm $f : E \to \{1, 2, \dots, k\}$ thỏa mãn điều kiện $f(e) \neq f(e')$ nếu e và e' liên thuộc với cùng một đỉnh. *Sắc số cạnh (edge chromatic number)* của một đồ thị G, ký hiệu $\chi'(G)$, là số màu nhỏ nhất có thể dùng để tô màu các cạnh của G

(a) Tìm $\chi'(C_n)$ và $\chi'(W_n)$ với $n \geq 3$

(b) Chứng minh rằng $\chi'(G) \geq \Delta(G)$, trong đó $\Delta(G)$ là bậc lớn nhất của một đỉnh của G

Gọi $\Delta(G)$ là *bậc lớn nhất của các đỉnh của đồ thị G*

Định lý 11

Cho $G = (V, E)$ là đơn đồ thị vô hướng có n đỉnh. Ta có $\chi(G) \leq \Delta(G) + 1$

Chứng minh.

Một thuật toán tham lam để tô màu các đỉnh của G bằng $\Delta(G) + 1$ màu $\{1, \dots, \Delta(G) + 1\}$ là như sau:

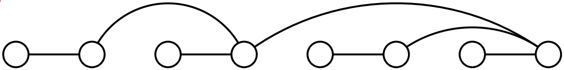
1. Gán nhãn $v_1, v_2, \dots, v_n$ cho các đỉnh của G một cách tùy ý
2. Với i từ 1 đến n, tô màu đỉnh v_i bằng màu nhỏ nhất trong số các màu chưa được tô cho bất kỳ đỉnh nào trong $N(v_i)$

Ta chứng minh rằng *Bước 2 của thuật toán luôn thực hiện được với $\Delta(G) + 1$ màu*. Thật vậy, đỉnh v_i có tối đa $\Delta(G)$ đỉnh kề với nó, do đó số màu tối đa sử dụng để tô màu các đỉnh trong $N(v_i)$ là $\Delta(G)$, nghĩa là luôn có ít nhất một trong số $\Delta(G) + 1$ màu không được sử dụng cho bất kỳ đỉnh nào kề với v_i, và ta có thể tô màu v_i bằng màu nhỏ nhất trong số các màu này $\square$

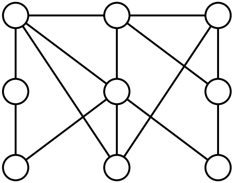
Bài tập 29

Sử dụng thuật toán ở Định lý 11 để tô màu các đồ thị sau

(a)



(b)



Trên thực tế, với phần lớn các đồ thị, chỉ cần $\Delta(G)$ màu là đủ

Định lý 12: Định lý Brook

Nếu G không phải là một chu trình độ dài lẻ hoặc một đồ thị đầy đủ thì $\chi(G) \leq \Delta(G)$

Bổ đề 13

Mọi đơn đồ thị phẳng và liên thông G gồm n đỉnh có một cách sắp thứ tự các đỉnh $v_1, v_2, \dots, v_n$ sao cho mỗi đỉnh kề với tối đa 5 đỉnh đứng trước nó

Chứng minh.

Ta chứng minh bằng quy nạp theo n.

- **Bước cơ sở:** Với $n \leq 6$, bất kể thứ tự sắp xếp các đỉnh nào đều thỏa mãn Bổ đề
- **Bước quy nạp:**
  - Giả sử Bổ đề đúng với mọi $6 \leq n \leq k$, trong đó $k \geq 6$ là một số nguyên nào đó. Ta chứng minh Bổ đề đúng với $n = k + 1$.
  - Thật vậy, giả sử G là đồ thị bất kỳ gồm $k + 1$ đỉnh. Từ Bài tập 25, tồn tại một đỉnh v của G thỏa mãn $\deg(v) \leq 5$.
  - Đồ thị $G - v$:
    - có tối đa 5 thành phần liên thông $G_1, G_2, \dots, G_5$
    - mỗi G_i có $n_i \leq k$ đỉnh $(1 \leq i \leq 5)$
    - và $n_1 + \cdots + n_5 = k$
  - Từ giả thiết quy nạp, tồn tại một thứ tự $v_1, \dots, v_k$ các đỉnh của $G - v$ thỏa mãn Bổ đề.
  - Đặt $v_{k+1} = v$, ta có $v_1, \dots, v_k, v_{k+1}$ là một thứ tự các đỉnh của G thỏa mãn Bổ đề

□

Lý thuyết đồ thị II

Hoàng Anh Đức

Đường đi Euler và Đường đi Hamilton
Đường đi Euler
Đường đi Hamilton

Bài toán đường đi ngắn nhất
Đồ thị có trọng số
Thuật toán Dijkstra

Đồ thị phẳng
Định nghĩa và khái niệm
Công thức Euler
Định lý Kuratowski

Tô màu đồ thị
Giới thiệu
Một số tính chất cơ bản
Tô màu đồ thị phẳng

Định lý 14

Mọi đồ thị phẳng G có $\chi(G) \leq 6$

Chứng minh.

Ta chỉ ra một cách tô màu đơn đồ thị phẳng và liên thông G bằng 6 màu

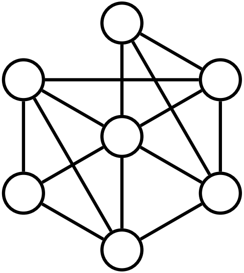
1. Tìm thứ tự các đỉnh $v_1, v_2, \dots, v_n$ của G thỏa mãn Bổ đề 13: mỗi đỉnh có tối đa 5 đỉnh kề đứng trước nó
2. Áp dụng thuật toán tham lam ở Định lý 11 với thứ tự đỉnh tìm được ở Bước 1

Chú ý rằng

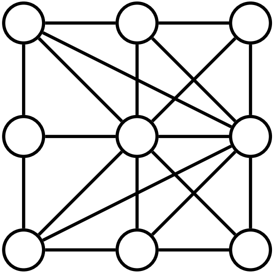
- Khuyên và cạnh song song (nếu có) không ảnh hưởng gì đến quá trình tô màu
- Nếu đồ thị phẳng đã cho không liên thông, ta có thể áp dụng quá trình tô màu riêng biệt cho từng thành phần liên thông

Bài tập 30

Sử dụng thuật toán ở Định lý 14 để tô màu các đồ thị phẳng sau bằng 6 màu $\{1, 2, 3, 4, 5, 6\}$



Đồ thị G



Đồ thị H

Nếu có thể, hãy tìm một cách tô màu các đồ thị trên bằng 5 màu hoặc ít hơn

Bài tập 31

Chứng minh Định lý 14 bằng phương pháp quy nạp

Định lý 15

Mọi đồ thị phẳng G có $\chi(G) \leq 5$

Bài tập 32

Bài tập sau đề xuất một cách chứng minh Định lý 15 bằng phương pháp quy nạp theo số đỉnh n của G

(a) Ở bước cơ sở, chứng minh Định lý 15 đúng với mọi $n \leq 5$

(b) Ở bước quy nạp, giả sử Định lý 15 đúng với mọi đồ thị phẳng gồm n đỉnh, trong đó $n \geq 5$ là số nguyên nào đó. Chứng minh Định lý 15 cũng đúng với mọi đồ thị phẳng gồm $n+1$ đỉnh. Giả sử G là đồ thị phẳng với $n+1$ đỉnh. Từ Bài tập 25, tồn tại một đỉnh v của G thỏa mãn $\deg(v) \leq 5$. Theo giả thiết quy nạp, có một cách tô màu α các đỉnh của $G - v$ bằng 5 màu $\{1, 2, 3, 4, 5\}$

Bài tập 32 (tiếp)

(b.1) Nếu $\deg(v) \leq 4$ hoặc $\deg(v) = 5$ và các đỉnh kề với v được tô màu bằng ≤ 4 màu, chứng minh rằng các đỉnh của G có thể được tô màu bằng 5 màu

(b.2) ($\star$) Nếu $\deg(v) = 5$ và các đỉnh kề với v, ví dụ v_1, v_2, v_3, v_4, v_5, được tô màu bằng 5 màu khác nhau, ví dụ như lần lượt là $1, 2, 3, 4, 5$, chứng minh rằng các đỉnh của G có thể được tô màu bằng 5 màu

- Xét đồ thị $G_{1,3}$ cảm sinh bởi các đỉnh tô màu 1 và 3. Điều gì xảy ra nếu v_1 và v_3 không thuộc cùng một thành phần liên thông của $G_{1,3}$? (**Gợi ý:** Trong trường hợp này, liệu có cách tô màu nào để v_1 và v_3 có cùng màu không?)
- Nếu v_1 và v_3 thuộc cùng một thành phần liên thông của $G_{1,3}$, xét đồ thị $G_{2,4}$ cảm sinh bởi các đỉnh tô màu 2 và 4. Liệu có thể xảy ra trường hợp v_2 và v_4 thuộc cùng một thành phần liên thông của $G_{2,4}$ không? (**Gợi ý:** G là đồ thị phẳng)

Part I

Phụ lục

Nội dung

Thuật toán Floyd:

- **Lịch sử:**
  - Được phát triển bởi Robert W. Floyd vào năm 1962 [Floyd 1962]
  - Tuy nhiên, nó về cơ bản giống với các thuật toán đã được công bố trước đó bởi Bernard Roy vào năm 1959 [Roy 1959] và cũng bởi Stephen Warshall vào năm 1962 [Warshall 1962] để tìm bao đóng bắc cầu (transitive closure) của đồ thị
- **Điều kiện:**
  - Trọng số có thể âm
  - *Không có chu trình có tổng trọng số âm*
- **Ý tưởng:**
  - Xét đường đi từ v_j đến v_k chỉ qua các đỉnh $\{v_0, v_1, \dots, v_i\}$
  - Nếu đường đi từ v_j đến v_k qua v_i ngắn hơn đường đi hiện tại, ta cập nhật khoảng cách

Thuật toán 2: Thuật toán Floyd [Floyd 1962]

Input: Đơn đồ thị vô hướng $G = (V, E, w)$ trong đó $V = \{v_0 = a, v_1, \dots, v_n = z\}$, $w : [V]^2 \to \mathbb{R} \cup \{\infty\}$ với $w(v_i, v_j) = \infty$ nếu $v_iv_j \notin E$, và G không có chu trình có tổng trọng số âm

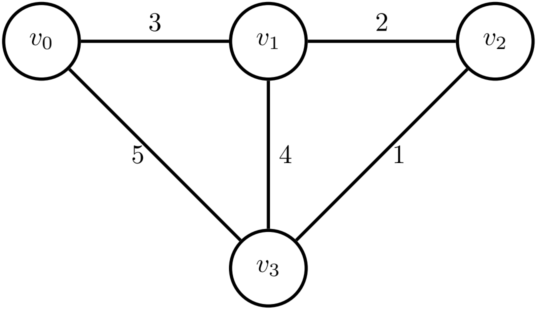
Output: Khoảng cách $d_G(v_i, v_j)$ với mọi $0 \leq i < j \leq n$

for $i := 0$ **to** n **do**
  for $j := 0$ **to** n **do**
    $d(v_i, v_j) := w(v_i, v_j)$

for $i := 0$ **to** n **do**
  for $j := 0$ **to** n **do**
    for $k := 0$ **to** n **do**
      if $d(v_j, v_i) + d(v_i, v_k) < d(v_j, v_k)$ **then**
        $d(v_j, v_k) := d(v_j, v_i) + d(v_i, v_k)$

return $\{d(v_i, v_j) \mid i, j \in \{0, 1, \dots, n\}\}$

Bài toán đường đi ngắn nhất
Thuật toán Floyd: Ví dụ



Hình: Đồ thị ví dụ cho thuật toán Floyd

Bài toán đường đi ngắn nhất
Thuật toán Floyd: Khởi tạo

- Khởi tạo ma trận khoảng cách $d(v_i, v_j) = w(v_i, v_j)$:

d	v_0	v_1	v_2	v_3
v_0	0	3	∞	5
v_1	3	0	2	4
v_2	∞	2	0	1
v_3	5	4	1	0

- Các giá trị ∞ cho biết không có cạnh trực tiếp nối hai đỉnh
- Thuật toán sẽ xét lần lượt các đỉnh trung gian i từ v_0 đến v_3

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Lặp với $i = 0$ (v_0)

- Xét v_0 làm đỉnh trung gian, kiểm tra $d(v_j, v_0) + d(v_0, v_k) < d(v_j, v_k)$?
- Không có cải thiện khoảng cách khi xét v_0 làm đỉnh trung gian

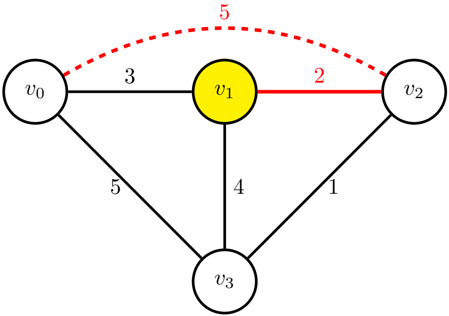
d	v_0	v_1	v_2	v_3
v_0	0	3	∞	5
v_1	3	0	2	4
v_2	∞	2	0	1
v_3	5	4	1	0

Bài toán đường đi ngắn nhất

Thuật toán Floyd: Lặp với $i = 1$ (v_1)

- Xét v_1 làm đỉnh trung gian, kiểm tra $d(v_j, v_1) + d(v_1, v_k) < d(v_j, v_k)$?
- Tìm được cải thiện: $d(v_0, v_1) + d(v_1, v_2) = 3 + 2 = 5 < \infty = d(v_0, v_2)$

d	v_0	v_1	v_2	v_3
v_0	0	3	5	5
v_1	3	0	2	4
v_2	5	2	0	1
v_3	5	4	1	0

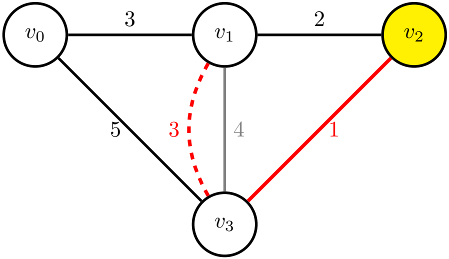


Bài toán đường đi ngắn nhất

Thuật toán Floyd: Lặp với $i = 2$ (v_2)

- Xét v_2 làm đỉnh trung gian, kiểm tra $d(v_j, v_2) + d(v_2, v_k) < d(v_j, v_k)$?
- Tìm được cải thiện: $d(v_1, v_2) + d(v_2, v_3) = 2 + 1 = 3 < d(v_1, v_3) = 4$

d	v_0	v_1	v_2	v_3
v_0	0	3	5	5
v_1	3	0	2	3
v_2	5	2	0	1
v_3	5	3	1	0



- Xét v_3 làm đỉnh trung gian, kiểm tra $d(v_j, v_3) + d(v_3, v_k) < d(v_j, v_k)$?
- Không có cải thiện khoảng cách khi xét v_3 làm đỉnh trung gian

d	v_0	v_1	v_2	v_3
v_0	0	3	5	5
v_1	3	0	2	3
v_2	5	2	0	1
v_3	5	3	1	0

- Ma trận khoảng cách cuối cùng cho kết quả khoảng cách giữa mọi cặp đỉnh

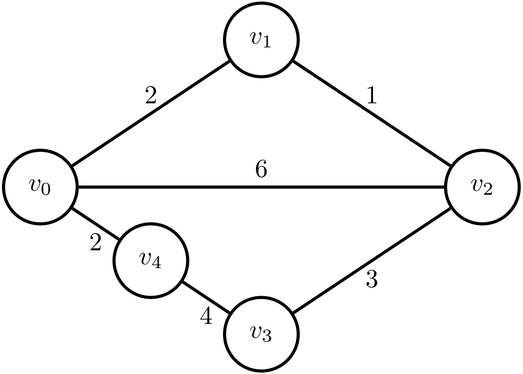
Bài toán đường đi ngắn nhất

Thuật toán Floyd: Phân tích

- **Tính đúng đắn:** [Floyd 1962]
- **Độ phức tạp**: $O(n^3)$ với $n = |V|$
- **Ưu điểm**:
  - Tìm khoảng cách giữa mọi cặp đỉnh
  - Có thể xử lý đồ thị có cạnh trọng số âm (không có chu trình âm)
- **Nhược điểm**:
  - Chậm hơn thuật toán Dijkstra khi chỉ cần tìm đường đi từ một nguồn
- Thuật toán có thể mở rộng để tìm đường đi ngắn nhất (không chỉ khoảng cách)

Bài tập 33

Áp dụng thuật toán Floyd để tìm khoảng cách giữa mọi cặp đỉnh trong đồ thị sau:



Một số lỗi thường gặp

Chú ý

Tham khảo từ tài liệu "Common Mistakes in Discrete Mathematics" (https://highered.mheducation.com/sites/dl/free/125967651x/1106131/Common_Mistakes_in_Discrete_Math.pdf)

(a) **Nhầm lẫn giữa định nghĩa đường đi và chu trình Hamilton và Euler**
- Hãy nhớ rằng có một phương pháp kiểm tra đơn giản cho đường đi và chu trình Euler, nhưng không có phương pháp kiểm tra đơn giản nào cho đường đi và chu trình Hamilton

(b) **Bỏ qua thực tế rằng việc có một chu trình Euler [Hamilton] hàm ý sự tồn tại của một đường đi Euler [Hamilton]**
- Hãy xem xét kỹ các định nghĩa

Một số lỗi thường gặp (tiếp)

(c) **Nhầm lẫn giữa các định lý trong lý thuyết đồ thị với mệnh đề đảo của chúng**

- Ví dụ, nếu trong một đồ thị đơn liên thông với $n \geq 3$ đỉnh, mỗi đỉnh có bậc ít nhất $n/2$, thì đồ thị đó có một chu trình Hamilton; nhưng mệnh đề đảo hoặc mệnh đề ngược của phát biểu này không đúng (có nhiều đồ thị có chu trình Hamilton mà trong đó bậc của các đỉnh nhỏ).
- Đây là một ví dụ khác: Nếu một đồ thị đơn liên thông là đồ thị phẳng, thì nó phải thỏa mãn $e \leq 3v - 6$, trong đó e là số cạnh và v là số đỉnh. Do đó (theo phản đảo), ta biết rằng nếu một đồ thị có quá nhiều cạnh ($e > 3v - 6$), thì nó không thể là đồ thị phẳng. Điều mà chúng ta không thể kết luận là mệnh đề đảo—không phải là một định lý—rằng nếu $e \leq 3v - 6$, thì đồ thị đó phải là đồ thị phẳng

(d) **Nhầm tưởng rằng một đồ thị không phẳng chỉ vì nó được vẽ theo cách mà có hai cạnh cắt nhau tại các điểm không phải là đầu mút**

Một số lỗi thường gặp (tiếp)

- Nếu có thể vẽ lại đồ thị đó mà không có cạnh nào cắt nhau ở các điểm không phải là đầu mút, thì đồ thị đó là đồ thị phẳng
- Ví dụ, K_4 là đồ thị phẳng, mặc dù khi vẽ nó với các đỉnh là đỉnh của một hình vuông và các đoạn thẳng biểu diễn các cạnh gây ra giao điểm ở giữa hình (Hãy vẽ lại nó dưới dạng các đỉnh của một tam giác với một đỉnh nữa ở bên trong)

(e) **Sai lầm khi kết luận rằng một khi đã tìm được cách tô màu đồ thị với n màu, thì số màu cần dùng của đồ thị đó phải là n**

- Thực tế, điều chúng ta biết trong trường hợp đó chỉ là số màu cần dùng nhiều nhất là n
- Có thể tìm được cách tô màu khác với ít hơn n màu
- Ví dụ, người ta có thể tô C_4 bằng bốn màu (một màu khác nhau cho mỗi đỉnh), nhưng số màu cần dùng thực sự là 2

(f) **Nhầm tưởng rằng thuật toán tham lam luôn đưa ra giải pháp tối ưu cho một bài toán**

Một số lỗi thường gặp (tiếp)

- Mặc dù trong nhiều trường hợp giải thuật tham lam đơn giản có thể tìm ra giải pháp tốt nhất (ví dụ, trong việc tìm kiếm cây khung nhỏ nhất), nhưng thường thì giải thuật tham lam hay thất bại (ví dụ, trong việc tìm cách tô màu đồ thị sử dụng ít màu nhất có thể)

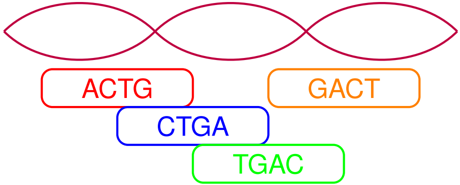
(g) **Không nhận ra rằng việc viết ra một thuật toán không đảm bảo rằng thuật toán đó làm những gì bạn muốn**

- Ví dụ, không thể viết một thuật toán tham lam để tô màu đồ thị và sau đó khẳng định mà không có sự chứng minh rằng thuật toán này tìm ra cách tô màu với số lượng màu ít nhất có thể

Đường đi Euler

Ứng dụng: Ghép các mảnh DNA

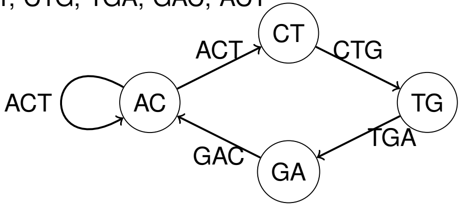
- **Bài toán giải trình tự DNA:** Xác định trình tự nucleotide trong một phân tử DNA
- **Thách thức:** Các thiết bị chỉ đọc được các đoạn ngắn (reads), cần ghép các đoạn này thành chuỗi hoàn chỉnh
- **Phương pháp:** Sử dụng đồ thị de Bruijn và đường đi Euler để tái tạo trình tự ban đầu



Hình: Đọc các đoạn mã di truyền ngắn từ chuỗi DNA

- k**-mer:** Các đoạn con có độ dài k của chuỗi DNA
- **Đồ thị de Bruijn:**
  - **Đỉnh:** Các $(k-1)$-mers
  - **Cạnh:** Từ $(k-1)$-mer u đến $(k-1)$-mer v nếu có k-mer trong đó u là tiền tố và v là hậu tố
  - **Nhãn cạnh:** k-mer tương ứng

Chuỗi ban đầu: ACTGACT
3-mers: ACT, CTG, TGA, GAC, ACT

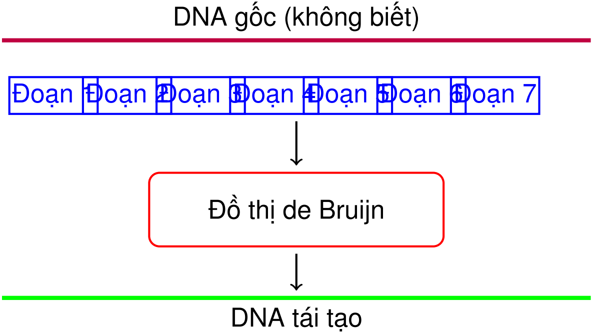


Hình: Đồ thị de Bruijn cho chuỗi ACTGACT với $k = 3$

Đường đi Euler

Tái tạo chuỗi DNA bằng đường đi Euler

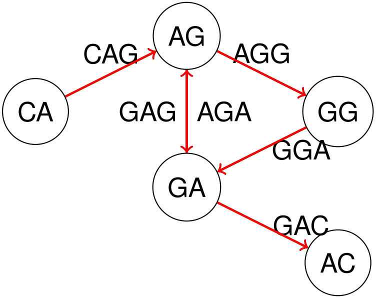
- **Lý thuyết:** Chuỗi DNA ban đầu tương ứng với một đường đi Euler trong đồ thị de Bruijn
- **Phương pháp:**
  - Tạo đồ thị de Bruijn từ tập các k-mers
  - Tìm đường đi Euler trong đồ thị
  - Tái tạo chuỗi từ đường đi tìm được



Hình: Quy trình tái tạo chuỗi DNA sử dụng đồ thị de Bruijn

Đường đi Euler
Ví dụ minh họa

- **Các đoạn đọc:**
  - CAGGA
  - AGGAG
  - GGAGA
  - GAGAC
- **3-mers:**
  - CAG, AGG, GGA
  - AGG, GGA, GAG
  - GGA, GAG, AGA
  - GAG, AGA, GAC



Hình: Đồ thị de Bruijn với $k = 3$

Đường đi Euler: CA $\to$ AG $\to$ GG $\to$ GA $\to$ AG $\to$ GA $\to$ AC

Chuỗi tái tạo: CAGGAGAC

Đường đi Euler

Ưu và nhược điểm của phương pháp

Ưu điểm:

- Tính toán hiệu quả
- Xử lý được dữ liệu lớn
- Cơ sở toán học chặt chẽ
- Phương pháp thông dụng trong các công cụ lắp ráp trình tự DNA

Thách thức:

- Xử lý lỗi trong dữ liệu đọc
- Các trình tự lặp lại
- Sự tồn tại của nhiều đường đi Euler khác nhau
- Cần thêm thông tin bổ sung để xác định đường đi chính xác

Đường đi Euler

Ứng dụng: Bài toán người đưa thư Trung Hoa

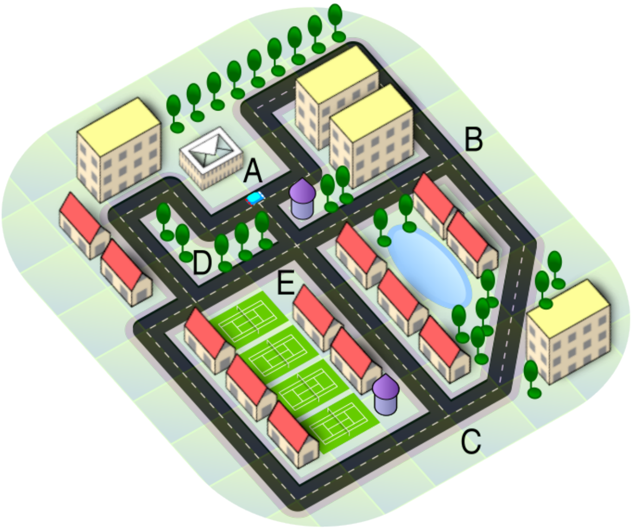
- **Bài toán người đưa thư Trung Hoa (Chinese Postman Problem - CPP):**
  - Đề xuất bởi nhà toán học Trung Quốc Mei-Ko Kwan (Quản Mai Cốc) năm 1962. Tên gọi của bài toán được đề xuất bởi Alan Goldman (Cục Tiêu chuẩn quốc gia Hoa Kỳ)
  - Mô hình hóa việc một người đưa thư cần đi qua tất cả các con đường trong khu vực
  - Người đưa thư muốn quay lại điểm xuất phát với quãng đường đi ngắn nhất có thể
- **Mối liên hệ với đường đi Euler:**
  - Nếu đồ thị có chu trình Euler: người đưa thư chỉ cần đi mỗi con đường đúng một lần
  - Nếu đồ thị không có chu trình Euler: người đưa thư phải đi qua một số con đường nhiều hơn một lần
- **Ứng dụng:** Tối ưu hóa lộ trình giao hàng, thu gom rác thải, bảo trì đường phố, v.v.

Đường đi Euler

Ứng dụng: Bài toán người đưa thư Trung Hoa

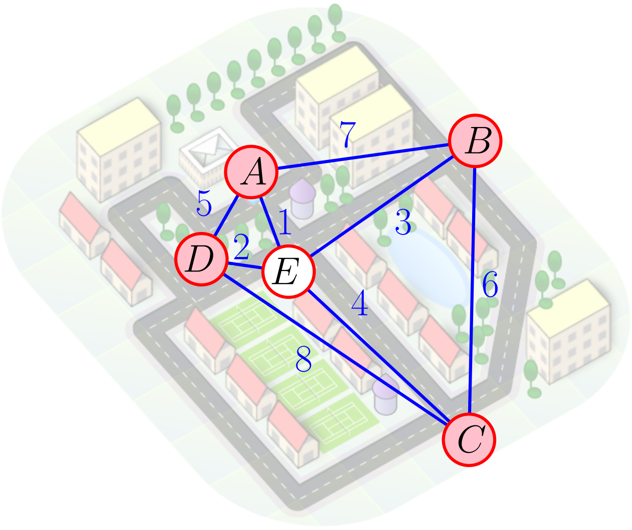
- **Đầu vào:** Đồ thị $G = (V, E)$ có trọng số (có thể có hướng hoặc vô hướng)
- **Đầu ra:** Chu trình với tổng trọng số nhỏ nhất đi qua mỗi cạnh ít nhất một lần
- **Giải thuật cho đồ thị vô hướng:**
  1. Kiểm tra tính liên thông của đồ thị. Nếu không liên thông, bài toán không có lời giải.
  2. Xác định tập O các đỉnh bậc lẻ trong đồ thị. ($|O|$ phải là số chẵn)
  3. Nếu $|O| = 0$ (không có đỉnh bậc lẻ):
     - Đồ thị đã có chu trình Euler, đây chính là lời giải tối ưu.
  4. Nếu $|O| > 0$ (có đỉnh bậc lẻ):
     - Với mỗi cặp đỉnh $u, v \in O$, tính khoảng cách $d(u, v)$ giữa chúng
     - Tìm bộ ghép hoàn hảo trọng số nhỏ nhất giữa các đỉnh trong O
     - Thêm các cạnh tương ứng với các đường đi ngắn nhất từ bộ ghép vào đồ thị ban đầu
  5. Tìm chu trình Euler trong đồ thị đã được bổ sung các cạnh và trả lại chu trình đó

Đường đi Euler
Ví dụ minh họa



Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

Đường đi Euler
Ví dụ minh họa



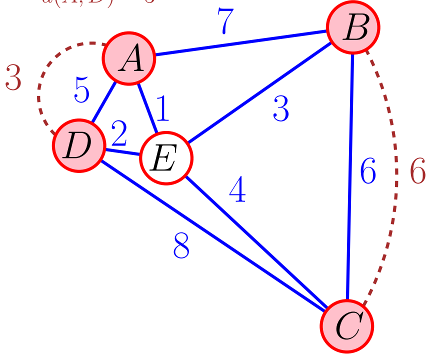
Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

Các đỉnh bậc lẻ: A, B, C, D

$d(A, B) = 4$ $d(B, C) = 6$ $d(C, D) = 6$

$d(A, C) = 5$ $d(B, D) = 5$

$d(A, D) = 3$



Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

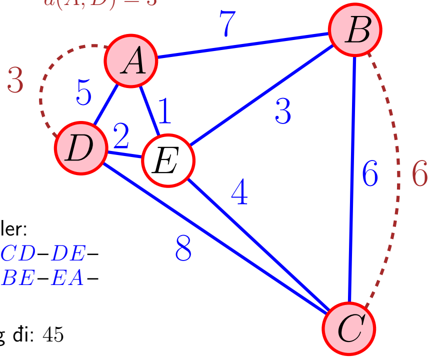
Đường đi Euler
Ví dụ minh họa

Các đỉnh bậc lẻ: A, B, C, D

$d(A,B) = 4$ $\quad d(B,C) = 6$ $\quad d(C,D) = 6$

$d(A,C) = 5$ $\quad d(B,D) = 5$

$d(A,D) = 3$



Chu trình Euler:
$AB–BC–CD–DE–EC–CB–BE–EA–AD–DA$

Độ dài đường đi: 45

Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

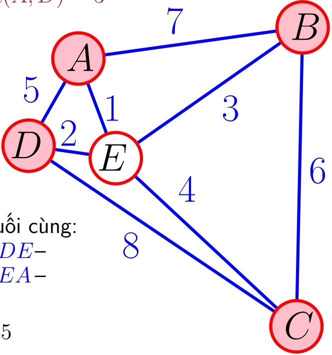
Đường đi Euler
Ví dụ minh họa

Các đỉnh bậc lẻ: A, B, C, D

$d(A,B) = 4$ $\quad d(B,C) = 6$ $\quad d(C,D) = 6$

$d(A,C) = 5$ $\quad d(B,D) = 5$

$d(A,D) = 3$



Chu trình trả lại cuối cùng:

$AB - BC - CD - DE -$
$EC - CB - BE - EA -$
$AD - DE - EA$

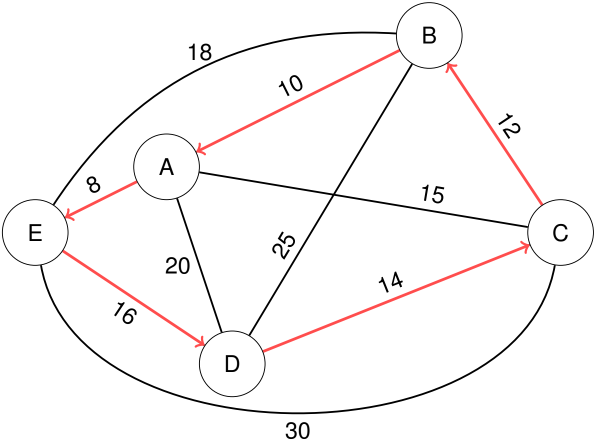
Độ dài đường đi: 45

Hình: Minh họa Bài toán người đưa thư Trung Hoa (ví dụ từ Wikipedia)

Đường đi Hamilton

Ứng dụng: Bài toán người giao hàng (TSP)

- **Bài toán người giao hàng (Traveling Salesman Problem - TSP):**
  - Người bán hàng cần thăm n thành phố, mỗi thành phố đúng một lần, rồi quay về nơi xuất phát
  - Mục tiêu: tìm hành trình có tổng chi phí (khoảng cách) nhỏ nhất
- **Mối liên hệ với đường đi Hamilton:**
  - Lời giải tối ưu của TSP chính là một chu trình Hamilton có trọng số nhỏ nhất
  - Đồ thị mô hình hóa bài toán là đồ thị đầy đủ có trọng số trên các cạnh
- **Bài toán NP-khó:** Không có thuật toán đa thức tìm nghiệm tối ưu (trừ khi P = NP)
- **Các ứng dụng trong thực tế:**
  - Tối ưu hóa lộ trình giao hàng
  - Lập kế hoạch sản xuất
  - Thiết kế mạch điện tử
  - Định tuyến phương tiện
  - Khoan đục bảng mạch (PCB)
  - Quy hoạch tour du lịch
  - Thu thập dữ liệu bằng drone



Chu trình tối ưu: A $\rightarrow$ E $\rightarrow$ D $\rightarrow$ C $\rightarrow$ B $\rightarrow$ A

Tổng chi phí: $8 + 16 + 14 + 12 + 10 = 60$

Hình: Ví dụ về bài toán TSP với 5 thành phố

Đường đi Hamilton

Ứng dụng: Mã Gray trong khoa học máy tính

- **Mã Gray (Gray Code):** Dãy các chuỗi nhị phân có độ dài n sao cho hai chuỗi liên tiếp chỉ khác nhau một bit
- **Mối liên hệ với đường đi Hamilton:**
  - Các mã Gray n-bit tương ứng với một đường đi Hamilton trong đồ thị siêu lập phương n chiều (n-cube)
  - Mỗi đỉnh là một chuỗi nhị phân độ dài n
  - Hai đỉnh kề nhau khi chuỗi nhị phân khác nhau đúng một bit
- **Ứng dụng trong lập trình và kỹ thuật số:**
  - Mã hóa vòng quay cho bộ mã hóa góc (encoder)
  - Tối thiểu hóa sai số trong chuyển đổi Analog-Digital
  - Tạo mã sửa lỗi và mã Hamming
  - Thuật toán backtracking và giải đệ quy

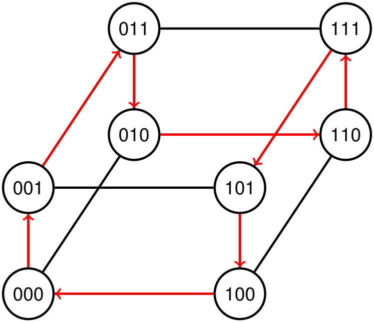
Mã Gray 3-bit:

- 000 (0)
- 001 (1)
- 011 (3)
- 010 (2)
- 110 (6)
- 111 (7)
- 101 (5)
- 100 (4)

$\rightarrow$ Quay về 000

Tạo mã Gray đệ quy:

1. Bắt đầu với 0 và 1 (Mã Gray 1-bit)
2. Xây dựng mã Gray n-bit từ mã Gray $(n-1)$-bit
   - Sao chép danh sách mã Gray $(n-1)$-bit
   - Thêm 0 vào trước nửa đầu danh sách
   - Thêm 1 vào trước nửa sau danh sách

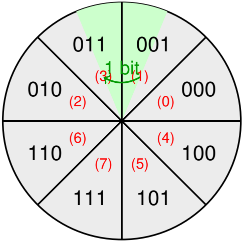


Hình: Khối 3 chiều Q_3. Các đỉnh là các mã Gray 3-bit. Đường đi Hamilton ứng với các mã Gray 3-bit được đánh dấu bằng mũi tên đỏ

Đường đi Hamilton

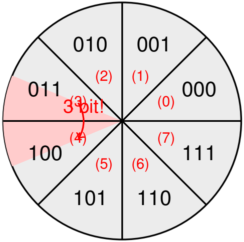
Ứng dụng của mã Gray trong kỹ thuật số

Mã Gray



001 → 011: **chỉ 1 bit thay đổi**

Mã nhị phân thông thường



011 → 100: **tất cả 3 bit đều thay đổi**

Ưu điểm của mã Gray: Chỉ một bit thay đổi giữa các vị trí kề nhau
⇒ *Giảm thiểu lỗi đọc khi bộ mã hóa ở vị trí trung gian*

Hình: So sánh bộ mã hóa vòng quay: Mã Gray (ít lỗi) và mã nhị phân thông thường (nhiều lỗi)

Đường đi Hamilton

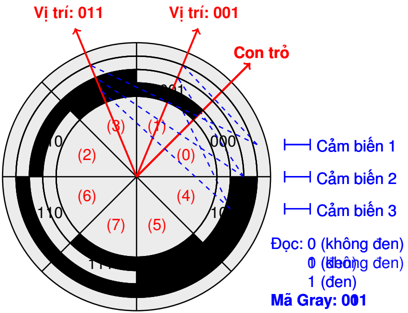
Ứng dụng của mã Gray trong kỹ thuật số

Bộ mã hóa vòng quay (rotary encoder):

- Chuyển đổi vị trí của con trỏ thành đầu ra số
- Sử dụng mã Gray để giảm thiểu lỗi khi đọc
- Khi sử dụng mã nhị phân thông thường, nhiều bit có thể thay đổi cùng một lúc

Ưu điểm của mã Gray:

- Chỉ một bit thay đổi mỗi lần
- Giảm thiểu sai số đọc (glitch)
- Tăng độ tin cậy của hệ thống



Chỉ bit giữa thay đổi!

Hình: Bộ mã hóa vòng quay sử dụng mã Gray 3-bit

Tài liệu tham khảo

Duan, Ran, Jiayi Mao, Xiao Mao, Xinkai Shu, and Longhui Yin (2025). "Breaking the Sorting Barrier for Directed Single-Source Shortest Paths". In: arXiv: 2504.17033.

Lewis, Rhyd M. R. (2021). *Guide to Graph Colouring: Algorithms and Applications*. 2nd. Springer. DOI: 10.1007/978-3-030-81054-2.

Rosen, Kenneth (2012). *Discrete Mathematics and Its Applications*. 7th. McGraw-Hill.

Robertson, Neil, Daniel Sanders, Paul Seymour, and Robin Thomas (1997). "The four-colour theorem". In: *Journal of Combinatorial Theory, Series B* 70.1, pp. 2–44. DOI: 10.1006/jctb.1997.1750.

Tài liệu tham khảo (tiếp)

Appel, Kenneth and Wolfgang Haken (1977). "Every planar map is four colorable. Part I: Discharging". In: *Illinois Journal of Mathematics* 21.3, pp. 429–490. DOI: `10.1215/ijm/1256049011`.

Appel, Kenneth, Wolfgang Haken, and John Koch (1977). "Every planar map is four colorable. Part II: Reducibility". In: *Illinois Journal of Mathematics* 21.3, pp. 491–567. DOI: `10.1215/ijm/1256049012`.

Hart, Peter E., Nils J. Nilsson, and Bertram Raphael (1968). "A formal basis for the heuristic determination of minimum cost paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2, pp. 100–107. DOI: `10.1109/TSSC.1968.300136`. This paper introduces the A* search algorithm.

Tài liệu tham khảo (tiếp)

- Floyd, Robert W. (1962). "Algorithm 97: Shortest Path". In: *Communications of the ACM* 5.6, p. 345. DOI: 10.1145/367766.368168.
- Warshall, Stephen (1962). "A theorem on Boolean matrices". In: *Journal of the ACM* 9.1, pp. 11–12. DOI: 10.1145/321105.321107.
- Dijkstra, Edsger W. (1959). "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1, pp. 269–271. DOI: 10.1007/BF01386390.
- Roy, Bernard (1959). "Transitivité et connexité". In: *C. R. Acad. Sci. Paris* 249, pp. 216–218. (In French.) This paper introduced what later became known as the Roy-Warshall algorithm for transitive closure.

Tài liệu tham khảo (tiếp)

- Fleury, M. (1883). "Deux problèmes de géométrie de situation". In: *Journal de Mathématiques Élémentaires*, pp. 257–261. (In French.) This paper contains Fleury's algorithm for finding Eulerian paths.
- Hierholzer, Carl and Christian Wiener (1873). "Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren". In: *Mathematische Annalen* 6.1, pp. 30–32. DOI: 10.1007/BF01442866. (In German.) This paper was published posthumously after Hierholzer's death and edited by Christian Wiener.
- Euler, Leonhard (1736). "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii Academiae Scientiarum Petropolitanae* 8, pp. 128–140. URL: https://scholarlycommons.pacific.edu/euler-works/53/. (In Latin.) English translation in Scientific American, 189 (1953), 66–70.