CHƯƠNG II
CÁC BÀI TOÁN VỀ ĐƯỜNG ĐI

I. ĐƯỜNG ĐI EULER

1. Chu trình, đường đi Euler

• ***Định nghĩa 2.1.1***

Cho đồ thị G=(V,E).

Chu trình Euler là chu trình qua mọi cạnh và mọi đỉnh đồ thị, mỗi cạnh không quá một lần.

Đường đi Euler là đường đi qua mọi cạnh và mọi đỉnh đồ thị, mỗi cạnh không quá một lần.

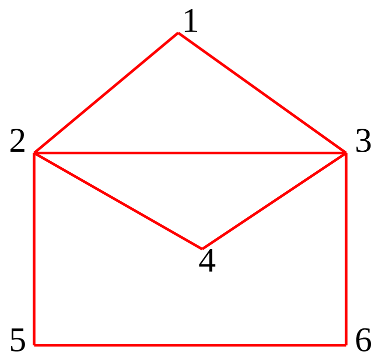
Cho đồ thị có hướng G=(V,E).

Chu trình có hướng Euler là chu trình có hướng qua mọi cung và mọi đỉnh đồ thị, mỗi cung không quá một lần.

Đường đi có hướng Euler là đường đi có hướng qua mọi cung và mọi đỉnh đồ thị, mỗi cung không quá một lần.

Đồ thị chứa chu trình Euler gọi là *Đồ thị Euler.*

◊ *Ví dụ 2.1.1.* Đồ thị

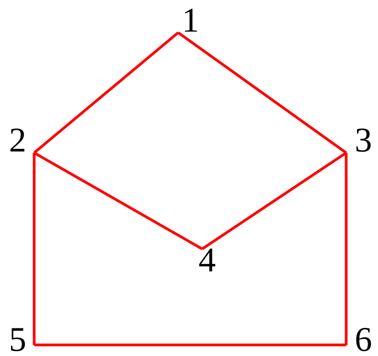


Hình 2.1.1.a

có chu trình Euler

1– 2– 4– 3– 6– 5– 2– 3– 1

Đồ thị



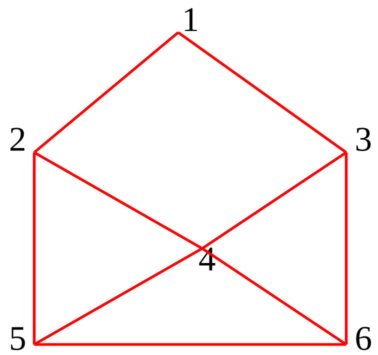
Hình 2.1.1.b

có đường đi Euler

2 – 5 – 6 – 3 – 4 – 2 – 1 –3

nhưng không có chu trình Euler.

Đồ thị



Hình 2.1.1.c

không có đường đi Euler.

2. Điều kiện cần và đủ

• ***Định lý 2.1.1*** (*Định lý Euler*)

Đồ thị G có chu trình Euler khi và chỉ khi G liên thông và mọi đỉnh có bậc chẵn lớn hơn 0.

Chứng minh

(i) (⇒): Giả sử G có chu trình Euler và v là đỉnh bất kỳ của G. Khi đó chu trình Euler đến v theo cạnh e thì ra khỏi v bằng cạnh e' ≠ e. Do đó bậc của v phải là số chẵn. G hiển nhiên là liên thông.

(ii) (⇐): Giả sử G liên thông và mọi đỉnh có bậc chẵn khác 0. Ta chứng minh G có chu trình Euler quy nạp theo số cạnh m của G.

* m = 1: Vì G liên thông và mọi đỉnh bậc chẵn nên G chỉ có 1 đỉnh và 1 khuyên. Khuyên đó cũng tạo thành chu trình Euler.

* Giả sử G có m cạnh, số đỉnh n > 0 và mọi đồ thị liên thông có số cạnh nhỏ hơn m với mọi đỉnh bậc chẵn đều có chu trình euler.

\+ Trường hợp n = 1 hoặc 2 thì hiển nhiên tồn tại chu trình Euler.

\+ Trường hợp n > 2. Vì bậc của các đỉnh chẵn ≥ 2, bao giờ cũng chọn được 3 đỉnh a, b, c với các cạnh x=(a,b), y=(a,c).

- Giả sử G chứa cạnh z=(b,c).

Xét đồ thị G' thu được từ G bằng cách loại bỏ ba cạnh x,y,z. Sẽ xảy ra 1 trong ba khả năng sau:

. G' liên thông. Vì số cạnh của G' nhỏ hơn m và các đỉnh vẫn có bậc chẵn nên theo giả thiết quy nạp tồn tại chu trình Euler C' của G'. Nối chu trình con (x,y,z) với C' ta thu được chu trình Euler C của G.

. G' có 2 thành phần liên thông G_1 và G_2. Không mất tính tổng quát giả sử G_1 chứa a, G_2 chứa b và c. G_1 có chu trình Euler C_1, G_2 có chu trình Euler C_2. Ta xây dựng chu trình Euler C của G như sau. Xuất phát từ đỉnh a đi theo chu trình C_1 quay về a, sau đó đi theo cạnh x=(a,b) đến đỉnh b, từ b đi theo chu trình C_2 quay về b, sau đó đi theo cạnh z=(b,c) và y=(c,a) quay về a.

. G’ có 3 thành phần liên thông G_1 , G_2 và G_3. Không mất tính tổng quát giả sử G_1 chứa a, G_2 chứa b và G_3 chứa c. G_1 có chu trình Euler C_1, G_2 có chu trình Euler C_2 , G_3 có chu trình Euler C_3. Ta xây dựng chu trình Euler C của G như sau. Xuất phát từ đỉnh a đi theo chu trình C_1 quay về a, sau đó đi theo cạnh x=(a,b) đến đỉnh b, từ b đi theo chu trình C_2 quay về b, sau đó đi theo cạnh z=(b,c) đến đỉnh c, từ c đi theo chu trình C_3 quay về c, sau đó đi theo cạnh y=(c,a) quay về a.

- Giả sử G không chứa cạnh z=(b,c)

Xét đồ thị G’ thu được từ G bằng cách loại bỏ 2 cạnh x,y và thêm cạnh z. Sẽ xảy ra 1 trong hai khả năng sau:

. G’ liên thông. Vì số cạnh của G’ nhỏ hơn m và các đỉnh vẫn có bậc chẵn nên theo giả thiết quy nạp tồn tại chu trình Euler C’ của G’. Thay cạnh $z \in$ C’ bằng cạnh x và y ta thu được chu trình Euler C của G.

. G’ có 2 thành phần liên thông G_1 và G_2. Không mất tính tổng quát giả sử G_1 chứa a, G_2 chứa b và c. G_1 có chu trình Euler C_1, G_2 có chu trình Euler C_2. Ta xây dựng chu trình Euler C của G như sau. Thay cạnh $z \in C_2$ bằng các cạnh x và y ta có chu trình C_2’. Nối C_2’ với C_1 ta thu được chu trình Euler C của G.

□

• ***Định lý 2.1.2***

Cho đồ thị G có 2*n* đỉnh bậc lẻ. Khi đó số đường đi tối thiểu phủ G là *n*.

Chứng minh.

Chứng minh quy nạp theo *n*.

(i) *n*=1: Nối 2 đỉnh bậc lẻ với nhau bằng cạnh z ta thu được đồ thị G’ thoả định lý Euler. Như vậy G’ có chu trình Euler C’. Bỏ cạnh z trên C’ ta thu được đường đi Euler phủ G.

(ii) Giả sử G có số đỉnh bậc lẻ là 2*n* và định lý đúng với (*n*–1). Nối 2 đỉnh bậc lẻ *a*,*b* nào đó với nhau bằng cạnh *z* ta thu được đồ thị G’ có 2*n*–2 đỉnh bậc lẻ. Theo giả thiết quy nạp G’ có *n*–1 đường đi phủ G’. Gọi P là đường đi qua cạnh *z*. Hiển nhiên *a*, *b* không phải đỉnh đầu hoặc cuối của P, vì vậy nếu bỏ cạnh *z* ta thu được 2 đường đi P_1 và P_2 cùng với *n*–2 đường đi còn lại phủ đồ thị G. □

◊ *Ghi chú*. Trong đồ thị có đỉnh bậc lẻ các đường đi tối thiểu phủ đồ thị phải có đỉnh đầu và đỉnh cuối là đỉnh bậc lẻ.

Bây giờ xét đồ thị có hướng G = (V, A). Ký hiệu

$$R = \{v \in V \mid \deg_I(v) = \deg_O(v)\}$$
$$S = \{v \in V \mid \deg_I(v) > \deg_O(v)\}$$
$$T = \{v \in V \mid \deg_I(v) < \deg_O(v)\}$$

Theo bổ đề bắt tay (Định lý 1.1.1) ta có

$$\sum_{v\in V}\deg_O(v) = \sum_{v\in V}\deg_I(v) \Rightarrow \sum_{v\in S}\left(\deg_I(v)-\deg_O(v)\right) = \sum_{v\in T}\left(\deg_O(v)-\deg_I(v)\right)$$

Ta ký hiệu

$$k = \sum_{v\in S}(\deg_I(v) - \deg_O(v)) = \sum_{v\in T}(\deg_O(v) - \deg_I(v))$$

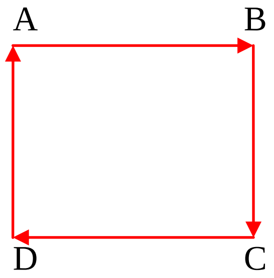
• ***Định lý* 2.1.3**

(i) Đồ thị có hướng G có chu trình có hướng Euler khi và chỉ khi G liên thông yếu và mọi đỉnh có nửa bậc vào bằng nửa bậc ra, tức S = ∅ và T = ∅.

(ii) Nếu S ≠ ∅, thì số đường đi có hướng tối thiểu phủ G là *k*. Các đường đi này nối các đỉnh của tập T đến các đỉnh của tập S.

Chứng minh. Tương tự như trường hợp vô hướng (bài tập).

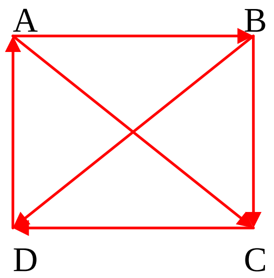
◊ *Ví dụ 2.1.2*. Đồ thị



Hình 2.1.2

có chu trình Euler: (A,B,C,D,A).

◊ *Ví dụ 2.1.3*. Đồ thị



Hình 2.1.3

không có chu trình Euler. Ta có

$$R = \{v \in V \mid \deg_I(v) = \deg_O(v)\} = \varnothing$$
$$S = \{v \in V \mid \deg_I(v) > \deg_O(v)\} = \{C, D\}$$
$$T = \{v \in V \mid \deg_I(v) < \deg_O(v)\} = \{A, B\}$$

và

$$k = \sum_{v\in S}(\deg_I(v) - \deg_O(v)) = \sum_{v\in T}(\deg_O(v) - \deg_I(v)) = 2$$

Vậy số đường đi có hướng tối thiểu phủ đồ thị là *k* = 2, ví dụ hai đường đi sau

(A→C→D→A→B→C) và (B→D)

3. Các thuật toán tìm chu trình Euler

• ***Thuật toán 2.1.1***

◊ *Đầu vào*. Đồ thị G ≠ ∅, không có đỉnh cô lập.

◊ *Đầu ra*. Chu trình Euler C của G, hoặc kết luận G không có chu trình Euler.

◊ *Phương pháp*

(0) Khởi tạo: Đặt H := G, k := 1, C := ∅, *v* ∈ G.

(1) Xuất phát từ *v*, xây dựng chu trình bất kỳ C_k trong H.

Nếu tồn tại C_k, nối C_k vào C, C := C ∪ C_k và sang bước (2).

Nếu không tồn tại C_k, thì kết luận *không có chu trình Euler*, kết thúc.

(2) Loại chu trình C_k khỏi H. Nếu H chứa các đỉnh cô lập, thì loại chúng khỏi H. Sang bước (3).

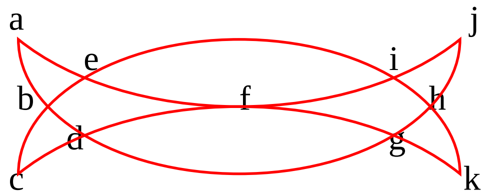
(3) Nếu H = ∅, thì kết luận C *là chu trình Euler*, kết thúc. Ngược lại, sang bước (4).

(4) Nếu H và C không có đỉnh chung, thì kết luận *không có chu trình Euler*, kết thúc.

Nếu H và C có đỉnh chung. Đặt k := k+1. Chọn đỉnh *v* ∈ C ∩ H bất kỳ. Quay lại bước (1).

◊ *Ví dụ 2.1.4*

Cho G là đồ thị *Thanh mã tấu Mohammed*.



Hình 2.1.4

Ta áp dụng thuật toán 2.1.1 để tìm chu trình Euler.

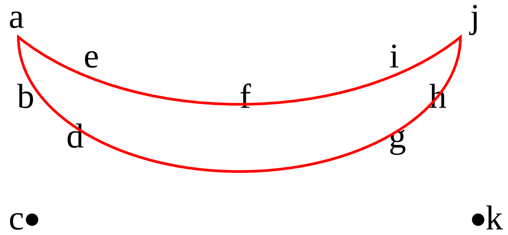
(0) Đặt H := G, k := 1, C := ∅, *v* := f.

(1) Ta xây dựng chu trình C_1 trong H:

$$C_1 := (f\text{-}g\text{-}k\text{-}h\text{-}i\text{-}e\text{-}b\text{-}c\text{-}d\text{-}f)$$

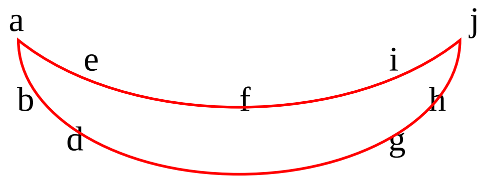
Đặt C := C ∪ C_1 = (f-g-k-h-i-e-b-c-d-f)

(2) Loại C_1 ra khỏi H, ta được đồ thị H như sau



Hình 2.1.5

Các đỉnh c và k là các đỉnh cô lập, vì thế ta loại chúng ra khỏi H và nhận được đồ thị H sau



Hình 2.1.6

(4) Chọn đỉnh chung của H và C là *v* := f. Đặt k := k+1 = 2. Quay lại bước (1)

(1) Ta xây dựng chu trình C_2 trong H:

$$C_2 := (f\text{-}i\text{-}j\text{-}h\text{-}g\text{-}d\text{-}b\text{-}a\text{-}e\text{-}f)$$

Nối C_2 vào C ta được chu trình C sau

$$C := C \cup C_2 = (f\text{-}g\text{-}k\text{-}h\text{-}i\text{-}e\text{-}b\text{-}c\text{-}d\text{-}f) \cup (f\text{-}i\text{-}j\text{-}h\text{-}g\text{-}d\text{-}b\text{-}a\text{-}e\text{-}f)$$

= (f-g-k-h-i-e-b-c-d-f-i-j-h-g-d-b-a-e-f)

(2) Loại C_2 ra khỏi H, ta được đồ thị H gồm toàn các đỉnh cô lập. Loại nốt các đỉnh cô lập ta có $H = \emptyset$.

(3) Vì $H = \emptyset$, ta kết luận *C là chu trình Euler*, kết thúc.

◊ *Ghi chú*. Hướng dẫn cài đặt thuật toán ở phần bài tập thực hành.

• *Thuật toán 2.1.2* (*Fleury*)

◊ Đầu vào. Đồ thị $G \neq \emptyset$, không có đỉnh cô lập.

◊ Đầu ra. Chu trình Euler C của G, hoặc kết luận G không có chu trình Euler.

◊ Phương pháp.

(1) Chọn đỉnh xuất phát bất kỳ v_0. Đặt $v_1 := v_0$, $C := (v_0)$. $H := G$.

(2) Nếu $H = \emptyset$, thì kết luận C là chu trình Euler, kết thúc. Ngược lại sang bước (3).

(3) Chọn cạnh đi tiếp:

Trường hợp đỉnh v_1 là đỉnh treo: Tồn tại duy nhất đỉnh v_2 kề v_1 .

Chọn cạnh (v_1 , v_2). Sang bước (4).

Trường hợp đỉnh v_1 không là đỉnh treo:

Nếu mọi cạnh liên thuộc v_1 là cầu, thì không có chu trình Euler, kết thúc.

Ngược lại, chọn cạnh (v_1, v_2) bất kỳ không phải là cầu trong H. Thêm vào đường đi C đỉnh v_2. Sang bước (4).

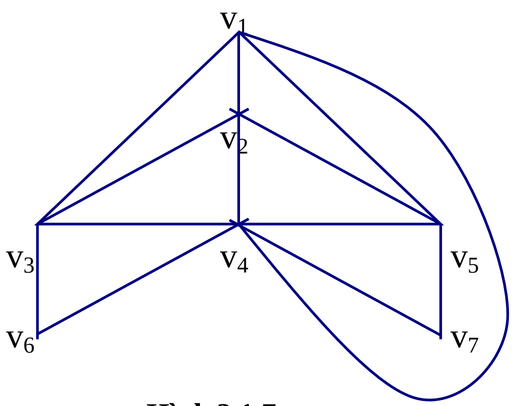
(4) Xoá cạnh vừa đi qua, và xoá đỉnh cô lập:

Loại khỏi H cạnh (v_1, v_2). Nếu H có đỉnh cô lập, thì loại chúng khỏi H.

Đặt $v_1 := v_2$. Sang bước (2).

◊ *Ví dụ 2.1.5*

Cho G là đồ thị hình sau



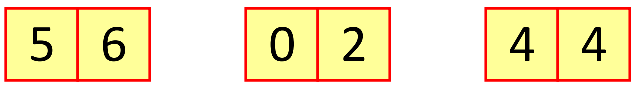
Hình 2.1.7

Đồ thị liên thông và có các đỉnh bậc chẵn. Ta có chu trình Euler sau

$(v_6 - v_4 - v_7 - v_5 - v_1 - v_3 - v_4 - v_2 - v_1 - v_4 - v_5 - v_2 - v_3 - v_6)$

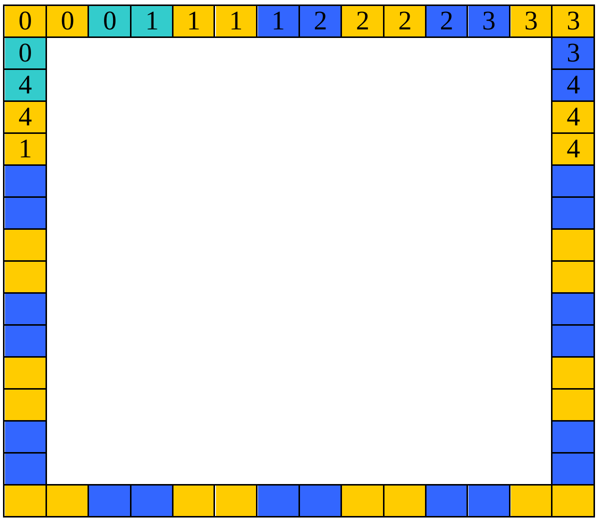
◊ *Ví dụ 2.1.6*

Domino là một hình chữ nhật chia thành 2 hình vuông mỗi hình mang một trong các số 0,1,2,3,4,5,6. Hai hình vuông trên một domino có thể mang cùng một số. Ví dụ



Hình 2.1.8

Có tất cả 28 quân Dômino khác nhau. Chứng minh rằng ta có thể sắp xếp các domino thành hình khép kín như hình dưới sao cho hai hình vuông kề nhau trên 2 domino khác nhau sẽ có cùng số (xem hình dưới).

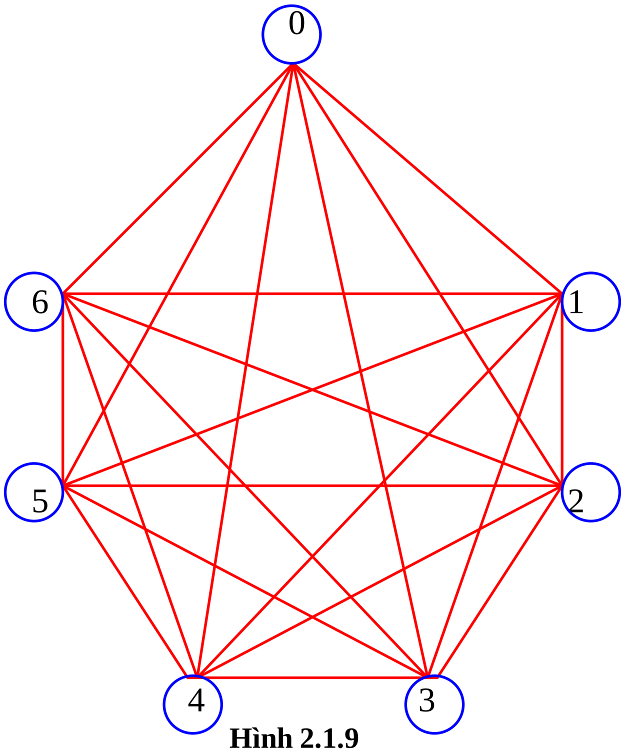


Giải

Ta lập đồ thị 7 đỉnh

0, 1, 2, 3, 4, 5, 6

Mỗi đỉnh có thể nối với các đỉnh còn lại và chính nó để tạo thành domino. Ta có đồ thị sau



Hình 2.1.9

liên thông với tất cả các đỉnh có bậc chẵn bằng 8. Do vậy tồn tại chu trình Euler. Mỗi chu trình Euler sẽ cho tương ứng một cách xếp.

◊ *Ví dụ 2.1.6. Bài toán người đưa thư.*

Nhân viên bưu điện nhận thư ở bưu cục, sau đó đi qua tất cả đường phố thuộc khối phố của mình phát thư rồi quay lại bưu cục. Biết rằng trong khối phố không có ngõ cụt, ngã 3, ngã 5, ngã 7, ... Tìm lộ trình ngắn nhất cho người đưa thư.

Vì người đưa thư phải đi qua tất cả đường phố nên mọi lộ trình đưa thư có độ dài không nhỏ hơn tổng độ dài tất cả đường phố. Theo giả thiết các điểm đường phố giao nhau có bậc chẵn, vì vậy tồn tại chu trình Euler. Vì chu trình Euler có độ dài bằng tổng độ dài tất cả đường phố, nên lộ trình đưa thư ngắn nhất chính là chu trình Euler.

Trường hợp đồ thị không có chu trình Euler sẽ nghiên cứu ở bài toán tìm đường đi ngắn nhất.

II. ĐƯỜNG ĐI TRONG MÊ CUNG

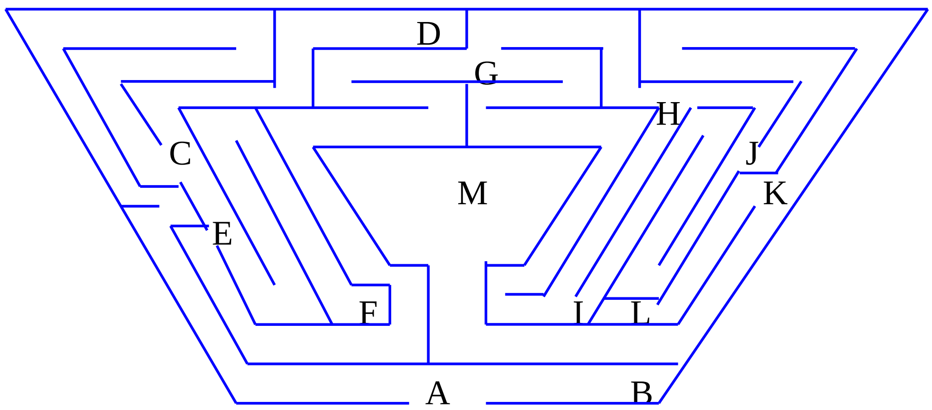
1. Phát biểu bài toán

Bài toán *tìm đường đi trong mê cung* là một trong các bài toán đố vui đồ thị lâu đời nhất. Một ví dụ trong văn học cổ Hi lạp là câu chuyện dũng sĩ *Theseus* đi cứu công chúa *Ariadne* bị con nhân mã *Minotaur* giam giữ trong mê cung.

Mê cung là hệ thống gồm nhiều hành lang nối với nhau. Bài toán *tìm đường đi trong mê cung* là đứng từ vị trí **s** (bên trong mê cung hoặc cửa vào) tìm đường đi đến vị trí **e** (cửa ra hoặc bên trong mê cung).

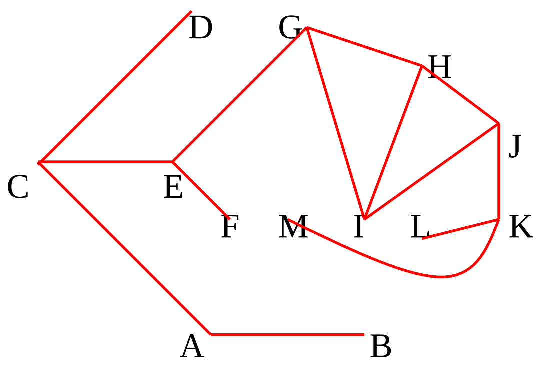
Nếu biểu diễn mê cung bằng đồ thị, trong đó các hành lang là cạnh, còn giao điểm của chúng là đỉnh thì ta có bài toán tìm đường đi trong đồ thị. Lưu ý rằng ta không biết trước sơ đồ của mê cung.

◊ *Ví dụ 2.2.1.* Cho mê cung



Hình 2.2.1

Mê cung trên có thể biểu diễn bằng đồ thị sau:



Hình 2.2.2

2. Thuật toán tìm đường đi trong mê cung

Cho đồ thị G=(V,E) và đỉnh $s, e \in$V. Tìm đường đi từ s đến e.

• ***Thuật toán 2.2.1*** (Wiener - 1873)

Xuất phát từ đỉnh **s** đi theo cạnh đồ thị theo nguyên tắc sau:

- Tại mỗi đỉnh chọn cạnh chưa đi qua trước đó.

- Nếu tại đỉnh nào đó mọi cạnh liên thuộc nó đã đi qua thì quay ngược lại cho đến khi gặp đỉnh có cạnh chưa qua.

Hiển nhiên là bằng cách này đi qua tất cả các cạnh của đồ thị. Tuy nhiên để có thể thực hiện thuật toán này cần phải nhớ thứ tự các cạnh đã đi qua, phải có phương tiện nhớ giống như "*cuộn chỉ Ariadne*". Thuật toán hiệu quả hơn là của Tarri sau đây.

• ***Thuật toán 2.2.2*** (*Tarri*)

Xuất phát từ đỉnh **s** đi theo (cạnh đồ thị) hành lang trong mê cung theo các nguyên tắc sau:

- *Đánh dấu* hướng đã đi qua của hành lang tại 2 đầu mút.

Với mỗi giao lộ (đỉnh) bậc ≥ 3 của mê cung (đồ thị), hành lang dẫn đến nó lần đầu tiên được *đánh dấu đặc biệt*.

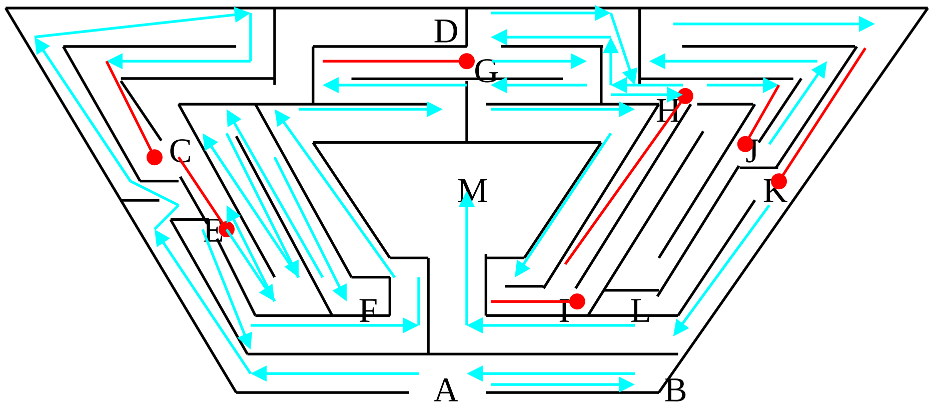
- *Chọn hành lang* (*cạnh*) *đi tiếp*: Tại mỗi đỉnh chọn hành lang chưa đi qua trước đó. Trường hợp tất cả hành lang đã đi qua thì chọn hành lang đi theo hướng ngược lại. Hành lang đánh dấu đặc biệt là phương án cuối cùng nếu không còn cách nào khác.

Bằng cách này ta đi qua tất cả các cạnh của đồ thị. Như vậy nếu đồ thị liên thông thì lúc nào đó ta sẽ đến đỉnh **e**.

◊ *Ví dụ 2.2.2*

Xét mê cung ở ví dụ trên. Ta tìm đường đi từ vị trí A (cổng) đến vị trí M.

Một phương án tìm đường đi cho ở sơ đồ dưới. Trong sơ đồ các mũi tên (xanh) chỉ hướng đi. Cạnh (đỏ) dẫn đến ngã ba, ngã tư được đánh dấu đặc biệt bằng núm tròn.



Hình 2.2.3

3. Một số bài toán ứng dụng

a. Bài toán sói, dê và cải

Một người cần chở *sói*, *dê* và *cải* qua sông bằng một thuyền nhỏ. Mỗi lần người chỉ chở được một thứ, hoặc sói hoặc dê hoặc cải và không được để sói đứng với dê hoặc dê đứng với cải mà không có người trông coi. Hãy chỉ cách chở.

Ký hiệu ***n*** (người), ***s*** (sói), ***d*** (dê) và ***c*** (cải). Ta lập đồ thị, biểu diễn khả năng chuyển đổi trạng thái người, sói, dê và cải ở bên bờ sông xuất phát. Mỗi nút trạng thái là tập con của (***nsdc***) trừ các tập (***sd***), (***nc***), (***dc***), (***ns***), **(*sdc*), (*n*)**. Sau đó áp dụng thuật toán trên tìm đường đi từ nút (***nsdc***) đến nút ∅.

Sau đây là hai phương án giải bài toán

$$\boldsymbol{nsdc \to sc \to nsc \to c \to ndc \to d \to nd \to \varnothing}$$

và

$$\boldsymbol{nsdc \to sc \to nsc \to s \to nsd \to d \to nd \to \varnothing}$$

b. Bài toán ba ông chồng ghen

Có ba cặp vợ chồng qua sông bằng một thuyền nhỏ. Mỗi lần thuyền chở được nhiều nhất 2 người và ai cũng biết bơi thuyền. Các ông chồng mắc bệnh ghen nặng nên không cho vợ đứng với người đàn ông khác khi không có mình. Hãy tìm phương án chở tất cả sang sông.

Ký hiệu các cặp chồng vợ là Aa, Bb, Cc. Ta lập đồ thị, biểu diễn khả năng chuyển đổi trạng thái các cặp chồng vợ ở bên bờ sông xuất phát. Mỗi nút trạng thái là tập con của (AaBbCc) trừ các tập dạng

$$\{S \mid ((aB) \subset S \text{ or } (aC) \subset S) \text{ and } A \not\in S\}$$
$$\{S \mid ((bA) \subset S \text{ or } (bC) \subset S) \text{ and } B \not\in S\}$$
$$\{S \mid ((cA) \subset S \text{ or } (cB) \subset S) \text{ and } C \not\in S\}$$

và các tập bù của chúng.

Sau đó áp dụng thuật toán trên để tìm đường đi từ nút AaBbCc đến nút ∅.

Sau đây là một phương án giải bài toán

$$\boldsymbol{AaBbCc \to AaBb \to AaBbC \to ABC \to ABCc \to Cc \to AaCc \to ac \to abc \to c \to Cc \to \varnothing}$$

c. Bài toán 3 thầy tu và 3 con quỷ

3 thầy tu và 3 con quỷ sang sông bằng 1 thuyền nhỏ. Mỗi lần thuyền chỉ chở được nhiều nhất 2 người và ai cũng biết bơi thuyền. Hãy tìm phương án sang sông sao cho nếu có cả thầy tu và quỷ trên một bờ sông thì số thầy tu không được ít hơn số quỷ (ngược lại thầy tu sẽ bị quỷ ăn thịt).

Ký hiệu nút trạng thái ở bờ sông xuất phát là (n,m), trong đó n là số thầy tu, m là số quỷ. Cặp (n,m) thỏa mãn

$$(0 \le n, m \le 3) \ \& \ [(n = 0) \text{ or } (n \ge m)] \ \& \ [(3-n = 0) \text{ or } (3-n \ge 3-m)]$$

Ta lập đồ thị, biểu diễn khả năng chuyển đổi trạng thái các thầy tu và quỷ ở bên bờ sông xuất phát.

Sau đó áp dụng thuật toán trên để tìm đường đi từ nút (3,3) đến (0,0).

Sau đây là một phương án giải bài toán

$$(3,3)\to(2,2)\to(3,2)\to(3,0)\to(3,1)\to(1,1)\to(2,2)\to(0,2)\to(0,3)\to(0,1)\to(1,1)\to(0,0)$$

III. ĐƯỜNG ĐI NGẮN NHẤT

1. Phát biểu bài toán

Cho đồ thị có trọng số G=(V,E,*w*). Ký hiệu *w*(i,j) là trọng số của cạnh (i,j). *Độ dài đường đi*

$$\mu = v_0 \to v_1 \to v_2 \to \dots \to v_{n-1} \to v_n$$

là tổng các trọng số

$$L(\mu) = \sum_{i=1}^{n} w(v_{i-1}, v_i)$$

• *Bài toán 1* (*1-1*). Cho hai đỉnh *a, z* của đồ thị. Bài toán đặt ra là tìm đường đi ngắn nhất từ *a* đến *z*.

• *Bài toán 2* (*1-n*). Cho đỉnh *a* của đồ thị. Bài toán đặt ra là tìm đường đi ngắn nhất từ *a* đến tất cả các đỉnh.

• *Bài toán 3* (*n-n*). Tìm đường đi ngắn nhất giữa mọi cặp đỉnh.

2. Thuật toán Dijkstra

Thuật giải tìm đường đi ngắn nhất từ đỉnh *a* đến đỉnh *z* trong đồ thị có trọng số. Trọng số của cạnh (i,j) là *w*(i,j) > 0 và đỉnh *x* sẽ mang nhãn L(*x*). Khi kết thúc thuật giải L(*z*) chính là chiều dài ngắn nhất từ *a* đến *z*.

• ***Thuật toán 2.3.1***

◊ *Đầu vào*. Đồ thị G=(V,E,w) có trọng số *w*(i,j) > 0 với mọi cạnh (i,j), đỉnh *a* và z.

◊ *Đầu ra*. L(z) chiều dài đường đi ngắn nhất từ *a* đến z, và đường đi ngắn nhất (nếu L(z) < +∞).

◊ *Phương pháp*

(1) *Khởi tạo*: Gán L(*a*):=0. Với mọi đỉnh $x \neq a$ gán L(*x*) := ∞. Đặt T := V.
Gán P(*x*) := ∅, ∀*x*∈V (P(*x*) là đỉnh trước đỉnh *x* trên đường đi ngắn nhất từ *a* đến *x*).

(2) Tính $m := \min\{L(u) \mid u \in T\}$.
Nếu *m* = +∞, kết luận không tồn tại đường đi từ *a* đến z. *kết thúc*.
Ngược lại, nếu *m* < +∞, chọn $v \in T$ sao cho L(*v*) = *m*, và đặt

$$T := T - \{v\}$$

Sang bước 3.

(3) Nếu z = *v*, L(z) là chiều dài đường đi ngắn nhất từ *a* đến z.
Từ z lần ngược theo đỉnh được ghi nhớ trong hàm P(*x*) ta tìm được đường đi ngắn nhất như sau:
Đặt $z_1 = P(z)$, $z_2 = P(z_1)$, …, $z_k = P(z_{k-1})$, $a = P(z_k)$.
Suy ra đường đi ngắn nhất là

$$a \to z_k \to z_{k-1} \to \dots \to z_1 \to z$$

Kết thúc.
Ngược lại, nếu $z \neq v$, sang bước (4).

(4) Với mỗi $x \in T$ kề (kề sau đối với đồ thị có hướng) *v*, tức là tồn tại cạnh (cung đối với đồ thị có hướng) (*v*,*x*), nếu

$$L(x) > L(v)+w(v,x),$$

thì gán

$$L(x) := L(v)+w(v,x)$$

và ghi nhớ đỉnh v cạnh đỉnh x, gán $P(x) := v$ (để sau này xây dựng đường đi ngắn nhất).

Quay về bước (2).

• ***Định lý 2.3.1.*** Thuật toán Dijkstra là đúng.

Chứng minh

Ký hiệu lần lượt các đỉnh v chọn ở bước (2) là $v_0 = a$, v_1, ... , $v_m = z$. Ta chứng minh bằng qui nạp rằng $L(v_i)$ chính là độ dài đường đi ngắn nhất từ a đến v_i , i=1,...,m.

- Bước cơ sở: Hiển nhiên $L(v_1)$ là độ dài ngắn nhất từ a đến v_1.

- Bước qui nạp: Giả thiết $L(v_i)$ chính là độ dài đường đi ngắn nhất từ a đến v_i với mọi $i < k$. Ta chứng minh rằng $L(v_k)$ là độ dài đường đi ngắn nhất từ a đến v_k.

Gọi P là đường đi ngắn nhất từ a đến v_k có độ dài l(P). Các đỉnh trên P trừ v_k phải thuộc

$$S = \{v_1, v_2,...,v_{k-1}\}.$$

Giả sử, ngược lại, gọi u là đỉnh đầu tiên trên P không thuộc S và v thuộc S là đỉnh trước u. Hiển nhiên

$$L(u) \le L(v) + w(v,u) < L(v_k) ,$$

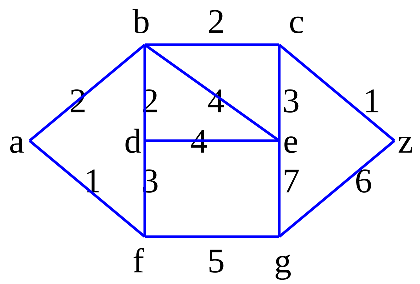
nên u phải bị loại khỏi T ở bước (2) trước v_k, tức u thuộc S, và đây là điều mâu thuẫn.

Bây giờ gọi v_h là đỉnh trước v_k trên P. Theo cách tính lại nhãn ta có

$$L(v_k) \le L(v_h) + w(v_h, v_k) = l(P).$$

Suy ra $L(v_k)$ là độ dài đường đi ngắn nhất từ a đến v_k. □

◊ *Ví dụ 2.3.1.* Tìm đường đi ngắn nhất từ đỉnh *a* đến đỉnh *z* trong đồ thị sau:



Hình 2.3.1

- Thực hiện bước 1:

Đặt

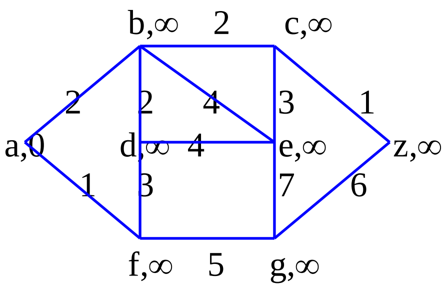
$$T := \{a,b,c,d,e,f,g,z\},$$

$$L(a) := 0,\ L(b)=L(c)=L(d)=L(e)=L(f)=L(g)=L(z):=\infty$$

và

$$P(a) = P(b)=P(c)=P(d)=P(e)=P(f)=P(g)=P(z):= \varnothing.$$

Các tham số được biểu diễn trên đồ thị như sau:



Hình 2.3.2

Các số sau đỉnh x là nhãn $L(x)$, $x\in T$.

- Thực hiện bước 2:

$$L(a) = \min\{L(x) \mid x\in T\}=0$$

suy ra $v = a$ và

$$T := T - \{a\} = \{b,c,d,e,f,g,z\}$$

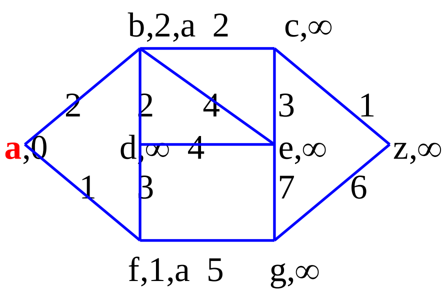
- Thực hiện bước 3: Vì $z \neq v$, sang bước 4.
- Thực hiện bước 4:

Xét đỉnh b và f kề đỉnh a. Ta có

$L(b) = \infty > L(a) + w(a,b) = 0 + 2 = 2 \Rightarrow L(b) := 2$, gán $P(b) := a$ (ghi nhớ đỉnh a cạnh đỉnh b).

$L(f) = \infty > L(a) + w(a,f) = 0 + 1 = 1 \Rightarrow L(f) := 1$, gán $P(f) := a$ (ghi nhớ đỉnh a cạnh đỉnh f).

Các đỉnh khác không thay đổi. Đồ thị có các nhãn như sau



Hình 2.3.3

- Thực hiện bước 2:

$$L(f) = \min\{L(x) \mid x\in T\}=1$$

suy ra $v = f$ và

$$T := T - \{f\} = \{b,c,d,e,g,z\}$$

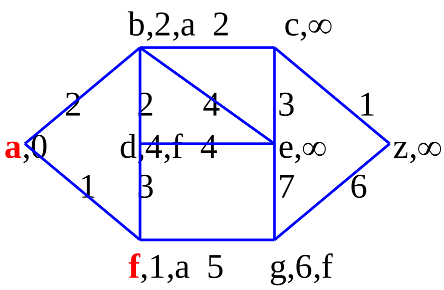
- Thực hiện bước 3: $z\neq v$, sang bước 4.
- Thực hiện bước 4:

Xét đỉnh d và g kề đỉnh f. Ta có

L(d) = ∞ > L(f) + w(f,d) = 1 + 3 = 4 ⇒ L(d) := 4, gán P(d):=f (ghi nhớ đỉnh f cạnh đỉnh d).

L(g) = ∞ > L(f) + w(f,g) = 1 + 5 = 6 ⇒ L(g) := 6, gán P(g):=f (ghi nhớ đỉnh f cạnh đỉnh g).

Các đỉnh khác không thay đổi. Đồ thị có nhãn như sau



Hình 2.3.4

- Thực hiện bước 2:

$$L(b) = \min\{L(x) \mid x \in T\} = 2$$

suy ra v = b và

$$T := T - \{b\} = \{c,d,e,g,z\}$$

- Thực hiện bước 3: z≠v, sang bước 4
- Thực hiện bước 4:

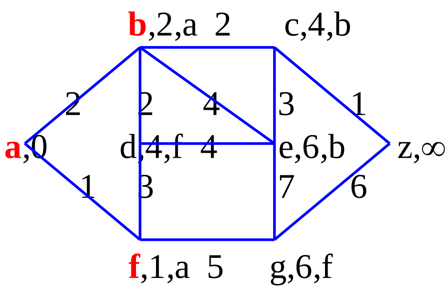
Xét đỉnh c, d và e kề đỉnh b. Ta có

L(c) = ∞ > L(b) + w(b,c) = 2 + 2 = 4 ⇒ L(c) := 4, P(c):=b (ghi nhớ đỉnh b cạnh đỉnh c).

L(d) = 4 = L(b) + w(b,d) = 2 + 2 ⇒ Nhãn đỉnh d không thay đổi.

L(e) = ∞ > L(b) + w(b,e) = 2 + 4 = 6 ⇒ L(e) := 6, P(e):=b (ghi nhớ đỉnh b cạnh đỉnh e).

Các đỉnh khác không thay đổi. Đồ thị có nhãn như sau



Hình 2.3.5

- Thực hiện bước 2:

$$L(c) = \min\{L(x) \mid x \in T\} = 4$$

suy ra v = c và

$$T := T - \{c\} = \{d,e,g,z\}$$

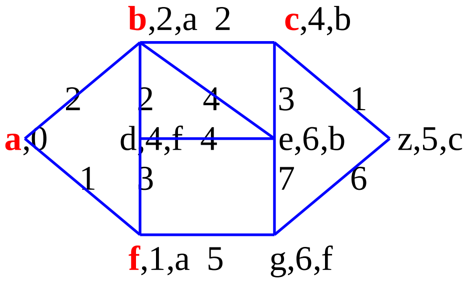
- Thực hiện bước 3: z≠v, sang bước 4
- Thực hiện bước 4:

Xét đỉnh e và z kề đỉnh c. Ta có

L(e) = 6 < L(c) + w(c,e) = 4 + 3 = 7 ⇒ Nhãn đỉnh e không thay đổi.

L(z) = ∞ > L(c) + w(c,z) = 4 + 1 = 5 ⇒ L(z) := 5, P(z):=c (ghi nhớ đỉnh c cạnh đỉnh z).

Các đỉnh khác không thay đổi. Đồ thị có nhãn như sau



Hình 2.3.6

- Thực hiện bước 2:

$$L(d) = \min\{L(x) \mid x \in T\} = 4$$

suy ra v = d và

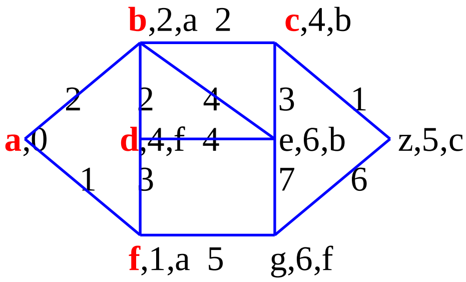
$$T := T - \{d\} = \{e,g,z\}$$

- Thực hiện bước 3: z≠v, sang bước 4
- Thực hiện bước 4:

Xét đỉnh e kề đỉnh d. Ta có

L(e) = 6 < L(d) + w(d,e) = 4 + 4 = 8 ⇒ Nhãn đỉnh e không thay đổi.

Các đỉnh khác không thay đổi. Đồ thị có nhãn như sau



Hình 2.3.7

- Thực hiện bước 2:

$$L(z) = \min\{L(x) \mid x \in T\} = 5$$

suy ra v = z

$$T = T - \{z\} = \{e,g\}$$

- Thực hiện bước 3: z = v, kết thúc.

L(z) = 5 là độ dài đường đi ngắn nhất từ a đến z.

Từ z ta đi ngược lại theo các đỉnh đã được ghi nhớ z→c→b→a:

$$P(z) = c,\ P(c) = b,\ P(b) = a.$$

Ta suy ra đường đi ngắn nhất là

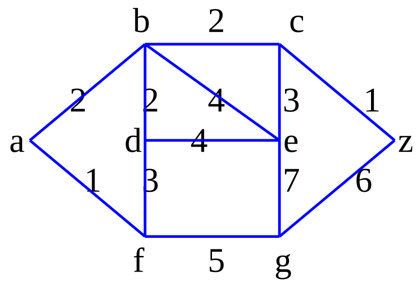
$$a \to b \to c \to z$$

• *Phương pháp lập bảng ghi nhãn*

Ta lập bảng tính toán các nhãn gồm các cột ứng với các đỉnh, và các hàng ứng với các lần tính nhãn ở bước (4). Các nhãn gạch dưới ứng với nhãn nhỏ nhất ở bước (2), và đỉnh bị loại được ghi bên phải.

Sau khi đỉnh z bị loại, từ z ta lần ngược về đỉnh a theo nhãn ghi trên bảng. Các đỉnh trên đường đi được gạch dưới (trên cột các đỉnh loại). Cuối cùng theo thứ tự ngược lại ta nhận được đường đi ngắn nhất.

Xét đồ thị hình 2.3.1



Hình 2.3.1

Sau đây là bảng tính toán nhãn của ví dụ trên

Vòng	a	b	c	d	e	f	g	z	Đỉnh loại
1:	$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞	$\boxed{a}$
2:	|	2(a)	∞	∞	∞	$\boxed{1}$(a)	∞	∞	f
3:	|	$\boxed{2}$	∞	4(f)	∞	|	6(f)	∞	$\boxed{b}$
4:	|	|	$\boxed{4}$(b)	4	6(b)	|	6	∞	$\boxed{c}$
5:	|	|	|	$\boxed{4}$	6	|	6	5(c)	d
6:	|	|	|	|	6	|	6	$\boxed{5}$	$\boxed{z}$

Ta suy ra đường đi ngắn nhất là

$$a \to b \to c \to z$$

có độ dài là 5.

• ***Định lý 2.3.2.*** Giả sử G là đồ thị liên thông có trọng số và có n đỉnh. Gọi f(n) là số lần thuật toán Dijkstra khảo sát một cạnh của G trong trường hợp xấu nhất. Khi đó ta có

$$f(n) = O(n^2)$$

Chứng minh

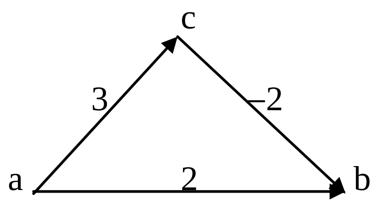
Thuật toán khảo sát các cạnh ở bước (4). Vì G có n đỉnh, nên số cạnh tối đa được khảo sát ở bước (4) là $n-1$. Bước (4) là vòng lặp chứa các bước (2)→(4). Vòng lặp thực hiện đến khi $z \notin T$. Vì ở mỗi vòng lặp ta rút ra một phần tử của T và khởi đầu T có n phần tử, nên vòng lặp được xử lý nhiều nhất n lần. Do đó

$$f(n) \le n(n-1) \le n^2.$$

◊ *Ghi chú*. Giả sử đồ thị K_n có trọng số. Bất kỳ thuật giải nào tìm đường đi ngắn nhất từ *a* đến *z* đều phải khảo sát mỗi cạnh ít nhất một lần. Vì K_n có $n(n-1)/2$ cạnh nên thuật giải phải khảo sát ít nhất $O(n^2)$ cạnh của K_n. Từ định lý trên ta thấy thuật toán Dijkstra là tối ưu.

3. Thuật toán Bellman-Ford

Thuật toán Dijkstra có thể không cho kết quả đúng nếu có trọng số âm. Xét đồ thị sau



Hình 2.3.8

Nếu áp dụng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh a, thì sau vòng lặp đầu tiên ta nhận được độ dài đường đi ngắn nhất từ a đến b là 2. Tuy nhiên, đường đi $a \to c \to b$ có độ dài là 1.

Thuật toán Bellman-Ford khắc phục hạn chế trên và cho phép tìm đường đi ngắn nhất từ một đỉnh đến các đỉnh khác mà không cần điều kiện trọng số dương. Thuật toán này cũng giúp xác định đồ thị có chu trình trọng số âm hay không.

• ***Thuật toán 2.3.2***

◊ *Đầu vào*. Đồ thị có trọng số (có hướng) $G=(V,E,w)$, $s \in V$.

◊ *Đầu ra*. Danh sách $L(v)$, độ dài đường đi (có hướng) ngắn nhất từ s đến v, và danh sách $P(v)$, đỉnh kề trước v trên đường đi ngắn nhất từ s đến v, $v \in V$, hoặc kết luận đồ thị có chu trình âm qua đỉnh khả nối với s.

◊ *Phương pháp*

(1) Khởi tạo: $L(s) := 0$; $L(v) := +\infty$, $\forall v \neq s$; $P(v) := \varnothing$, $\forall v \in V$.

(2) For i:= 1 to $n-1$ do ($n = |V|$ số phần tử của V)

For $(u,v) \in E$ do

if $L(v) > L(u) + w(u,v)$ then

$\{L(v) := L(u) + w(u,v);\ P(v) := u;\}$

(3) Nếu tồn tại $(u,v) \in E$ thỏa $L(v) > L(u) + w(u,v)$, thì kết luận đồ thị có chu trình âm khả nối với s.

Ngược lại, $L(v)$ là độ dài đường đi (có hướng) ngắn nhất từ s đến v, và danh sách $P(v)$, đỉnh kề trước v trên đường đi ngắn nhất từ s đến v, $v \in V$.

◊ *Phương pháp xác định đường đi ngắn nhất từ s đến v*

Đặt

$v_0 := v$; $v_1 := P(v_0)$; $v_2 := P(v_1)$; ... ; $v_i := P(v_{i-1})$; ... ; $v_k := P(v_{k-1}) = s$

Suy ra đường đi ngắn nhất từ s đến v là

$$s = v_k \to v_{k-1} \to \ldots \to v_1 \to v_0 = v$$

• ***Định lý 2.3.3.*** Thuật toán Bellman-Ford là đúng và có độ phức tạp $O(|V|.|E|)$.

Chứng minh

Ký hiệu $L_i(v)$, $\forall v \in V$, là giá trị $L(v)$ sau vòng lặp thứ i, $i=1,\ldots,|V|-1$, ở bước 2. Trước tiên ta chứng minh bổ đề sau quy nạp theo i.

◊ *Bổ đề*. Với mọi $v \in V$, $i=1,\ldots, |V|-1$, $L_i(v)$ là độ dài ngắn nhất trong số các đường đi từ s đến v có số cung không quá i.

Bổ đề hiển nhiên đúng với i = 1. Giả sử bổ đề đúng với 1, 2, ..., $i-1$. Ta chứng minh nó cũng đúng với i. Cho $v \in V$, ký hiệu μ là đường đi ngắn nhất trong số các đường đi từ s đến v có số cung không quá i và $L(\mu)$ là độ dài của μ. Ta xét hai trường hợp sau.

o Trường hợp μ có số cung nhỏ hơn i: μ cũng là đường đi ngắn nhất trong số các đường đi từ s đến v có số cung không quá i-1 và từ giả thiết quy nạp suy ra $L(\mu) = L_{i-1}(v) \geq L_i(v)$. Như vậy $L_i(v) = L(\mu)$ là độ dài ngắn nhất trong số các đường đi từ s đến v có số cung không quá i.

o Trường hợp μ có số cung bằng i: Ký hiệu (u,v) là cung cuối cùng trên đường đi μ và μ' là đường đi trên μ từ s đến u. Như vậy μ' có i-1 cung và là đường đi ngắn nhất trong số các đường đi từ s đến u có số cung không quá i-1 (vì nếu có đường đi μ'' từ s đến u ngắn hơn μ', thì đường đi $\mu'' \cup (u,v)$ sẽ ngắn hơn μ, mâu thuẫn). Theo giả thiết quy nạp $L(\mu') = L_{i-1}(u)$ và từ đó suy ra $L(\mu) = L(\mu') + w(u,v) = L_{i-1}(u) + w(u,v)$. Sau vòng lặp thứ i ta có $L_i(v) \leq L_{i-1}(u) + w(u,v) = L(\mu)$. Suy ra $L_i(v) = L(\mu)$ là độ dài ngắn nhất trong số các đường đi từ s đến v có số cung không quá i.

Bổ đề đã được chứng minh.

Theo bổ đề, sau vòng lặp, L(v) là độ dài đường đi (có hướng) ngắn nhất từ s đến v có số cung không quá $|V|-1$, $\forall v \in V$.

Giả sử tồn tại $(u,v) \in E$ thỏa $L(v) > L(u) + w(u,v) = L'(v)$. Ký hiệu μ_u là đường đi từ s đến u, μ_v là đường đi từ s đến v. Khi đó đường đi

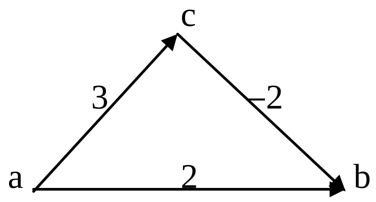
$$\mu' := \mu_u \cup (u,v)$$

là đường đi từ s đến v có độ dài $L'(v) < L(v)$. Vậy μ' phải có ít nhất $|V|$ cung, suy ra μ' chứa chu trình. Ký hiệu w là đỉnh lặp lại trên μ'. Giá trị L(w) Ở lần xuất hiện sau trên μ' phải nhỏ hơn ở lần xuất hiện trước. Suy ra chu trình có trọng số âm.

Độ phức tạp $O(|V|.|E|)$ suy ra từ việc thực hiện $|V|$ vòng lặp với mỗi vòng lặp thực hiện $|E|$ phép toán.

□

◊ *Ví dụ*: Xét đồ thị hình 2.3.8



Hình 2.3.8

Áp dụng thuật toán Bellman-Ford và ở bước 2 duyệt các cung theo thứ tự (a,b), (c,b), (a,c) ta được kết quả sau:

	L(a)	L(b)	P(b)	L(c)	P(c)
bước 1:	0	∞	∅	∞	∅
bước 2:	0	2	a	3	a
	0	1	c	3	a

bước 3: không tồn tại cung (u,v) thỏa $L(v) > L(u) + w(u,v)$.

Kết luận:

Độ dài đường đi ngắn nhất từ a đến b là d(a,b) = 1, từ a đến c là d(a,c) = 3.

Để tìm đường đi ngắn nhất từ a đến b ta làm như sau: Ta có

$$P(b) = c, P(c) = a$$

suy ra đường đi ngắn nhất là: $a \to c \to b$.

4. Thuật toán Floyd

Thuật giải tìm độ dài đường đi ngắn nhất giữa mọi cặp đỉnh trong đồ thị có hướng có trọng số (không bắt buộc ≥ 0).

• ***Thuật toán 2.3.3***

◊ *Đầu vào*. Đồ thị $G=(V,E)$, $V = \{1, 2, ... , n\}$, có trọng số $w(i,j)$ với mọi cung (i,j).

◊ *Đầu ra*. Ma trận $D=[d(i,j)]$, trong đó $d(i,j)$ là chiều dài đường đi ngắn nhất từ i đến j với mọi cặp (i,j).

◊ *Phương pháp*:

(1) Bước khởi tạo: Ký hiệu D_0 là ma trận xuất phát

$$D_0 = [d_0(i,j)]$$

trong đó $d_0(i,j) = w(i,j)$ nếu tồn tại cung (i,j) và $d_0(i,j) = +\infty$ nếu không tồn tại cung (i,j) (đặc biệt nếu không có khuyên tại i thì $d_0(i,i) = +\infty$).

Gán k:=0.

(2) Kiểm tra kết thúc: Nếu $k = n$, kết thúc. $D = D_n$ là ma trận độ dài đường đi ngắn nhất. Ngược lại tăng k lên 1 đơn vị (k:=k+1) và sang (3).

(3) Tính ma trận D_k theo D_{k-1} :

Với mọi cặp (i,j), i=1..n, j=1..n thực hiện:

Nếu $d_{k-1}(i,j) > d_{k-1}(i,k) + d_{k-1}(k,j)$ thì đặt

$$d_k(i,j) := d_{k-1}(i,k) + d_{k-1}(k,j)$$

ngược lại đặt

$$d_k(i,j) := d_{k-1}(i,j)$$

Quay lại bước (2).

• ***Định lý 2.3.4.*** Thuật toán Floyd là đúng.

Chứng minh.

Ta chứng minh, bằng qui nạp theo k, mệnh đề sau:

$d_k(i,j)$ là chiều dài đường đi ngắn nhất trong những đường nối đỉnh i đến j qua các đỉnh trung gian $\{1,2,...,k\}$.

- Bước cơ sở: Hiển nhiên mệnh đề đúng với k=0.

- Bước qui nạp: Giả sử mệnh đề đúng với k-1. Xét $d_k(i,j)$.

Sẽ xảy ra một trong hai trường hợp sau:

(i) Trong các đường nối đỉnh i với j qua các đỉnh trung gian $\{1,2,...,k-1,k\}$ có chiều dài ngắn nhất, tồn tại một đường **p** không qua đỉnh k. Khi đó p cũng là đường ngắn nhất nối đỉnh i với j qua các đỉnh trung gian $\{1,2,...,k-1\}$, nên theo giả thiết qui nạp

$$d_{k-1}(i,j) = d(p) \leq d_{k-1}(i,k) + d_{k-1}(k,j)$$

Do đó theo cách tính d_k ta có $d_k(i,j) = d_{k-1}(i,j) = d(p)$ là độ dài đường đi ngắn nhất từ i đến j qua các đỉnh trung gian $\{1,2,...,k-1,k\}$.

(ii) Mọi đường nối đỉnh i với j qua các đỉnh trung gian {1,2,...,k-1,k} có chiều dài ngắn nhất đều qua đỉnh k. Gọi p = (i,...,k,...,j) là một đường ngắn nhất nối đỉnh i với j qua các đỉnh trung gian {1,2,...,k-1,k}. Khi đó đoạn (i,...k) và (k,...,j) cũng phải là các đường đi ngắn nhất qua các đỉnh trung gian {1,2,...,k-1}. Ta có

$$d_{k-1}(i,k) + d_{k-1}(k,j) = d(p) < d_{k-1}(i,j)$$

(Bất đẳng thức cuối cùng suy ra từ giả thiết mọi đường đi nối đỉnh i với j qua các đỉnh trung gian {1,2,...,k-1,k} có chiều dài ngắn nhất đều qua đỉnh k).

Do đó theo cách tính d_k ta có $d_k(i,j) = d_{k-1}(i,k) + d_{k-1}(k,j) = d(p)$ là độ dài đường đi ngắn nhất từ i đến j qua các đỉnh trung gian {1,2,...,k-1,k}. □

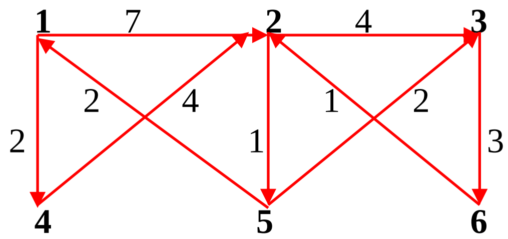
• ***Hệ quả 2.3.5***

(i) Nếu ma trận kết quả của thuật toán Floyd có phần tử hữu hạn trên đường chéo i=i thì đồ thị chứa chu trình.

(ii) Nếu ma trận kết quả chứa phần tử +∞ ngoài đường chéo i=i thì đồ thị không liên thông mạnh.

◊ *Ghi chú*. Từ hệ quả trên ta có thể sử dụng thuật toán Floyd, với w(i,j)=1 nếu tồn tại cung (i,j) và w(i,j)= +∞ nếu không tồn tại cung (i,j), để xác định xem đồ thị có chu trình hay có liên thông hay không.

◊ *Ví dụ 2.3.2*. Xét đồ thị sau



Hình 2.3.9

Áp dụng thuật toán Floyd ta có:

Ma trận khoảng cách xuất phát D_0 là (các ô trống là ∞):

D_0 =

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**
1		7		2		
2			4		1	
3						3
4		4				
5	2		2			
6		1				

Từ ma trận D_0 , theo thuật toán, ta xây dựng các ma trận tiếp theo như sau (các ô gạch dưới có giá trị thay đổi)

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**

$D_1 =$

1		7		2		
2			4		1	
3						3
4		4				
5	2	$\underline{9}$	2	$\underline{4}$		
6		1				

$D_2 =$

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**
1		7	$\underline{11}$	2	$\underline{8}$	
2			4		1	
3						3
4		4	$\underline{8}$		$\underline{5}$	
5	2	9	2	4	$\underline{10}$	
6		1	$\underline{5}$		$\underline{2}$	

$D_3 =$

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**
1		7	11	2	8	$\underline{14}$
2			4		1	$\underline{7}$
3						3
4		4	8		5	$\underline{11}$
5	2	9	2	4	10	$\underline{5}$
6		1	5		2	$\underline{8}$

$D_4 =$

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**
1		$\underline{6}$	$\underline{10}$	2	$\underline{7}$	$\underline{13}$
2			4		1	7
3						3
4		4	8		5	11
5	2	$\underline{8}$	2	4	$\underline{9}$	5
6		1	5		2	8

$D_5 =$

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**
1	$\underline{9}$	6	$\underline{9}$	2	7	$\underline{12}$
2	$\underline{3}$	$\underline{9}$	$\underline{3}$	$\underline{5}$	1	$\underline{6}$
3						3
4	$\underline{7}$	4	$\underline{7}$	$\underline{9}$	5	$\underline{10}$
5	2	8	2	4	9	5
6	$\underline{4}$	1	$\underline{4}$	$\underline{6}$	2	$\underline{7}$

$D = D_6 =$

Đỉnh	**1**	**2**	**3**	**4**	**5**	**6**
1	9	6	9	2	7	12
2	3	$\underline{7}$	3	5	1	6
3	$\underline{7}$	$\underline{4}$	$\underline{7}$	$\underline{9}$	$\underline{5}$	3

4	7	4	7	9	5	10
5	2	<u>6</u>	2	4	<u>7</u>	5
6	4	1	4	6	2	7

Cuối cùng, D là ma trận khoảng cách ngắn nhất giữa các đỉnh. Theo hệ quả ta thấy đồ thị liên thông mạnh và chứa chu trình.

5. Thuật toán Floyd-Warshall

Thuật giải tìm đường đi ngắn nhất giữa mọi cặp đỉnh trong đồ thị (có hướng) có trọng số.

• ***Thuật toán 2.3.4***

◊ *Đầu vào*. Đồ thị $G=(V,E,w)$, $V= \{v_1, v_2, ... , v_n\}$, có trọng số $w(v_i,v_j)$ với mọi cạnh (cung) $(v_i,v_j)\in E$.

◊ *Đầu ra*. Ma trận $D=[d(v_i,v_j)]_{nxn}$, trong đó $d(v_i,v_j)$ là chiều dài đường đi ngắn nhất từ v_i đến v_j với mọi cặp (v_i,v_j).

Ma trận $P=[p(v_i,v_j)]_{nxn}$ dùng để xác định đường đi ngắn nhất.

◊ *Phương pháp*

(1) Bước khởi tạo: Ký hiệu D_0 là ma trận xuất phát

$$D_0 = [d_0(v_i,v_j)]$$

trong đó $d_0(v_i,v_j) = w(v_i,v_j)$ nếu tồn tại cung (v_i,v_j) và $d_0(v_i,v_j) = +\infty$ nếu không tồn tại cung (v_i,v_j) (đặc biệt nếu không có khuyên tại *i* thì $d_0(v_i,v_i) = +\infty$).

$$P_0 = [p_0(v_i,v_j)]$$

trong đó $p_0(v_i,v_j) = v_j$ nếu có cung từ v_i đến v_j và $p_0(v_i,v_j)$ không xác định nếu không có cung từ v_i đến v_j.

(2) For k:=1 to n do

Tính ma trận D_k và P_k theo D_{k-1} và P_{k-1}:

Với mọi cặp (v_i,v_j), i=1..n, j=1..n thực hiện:

Nếu $d_{k-1}(v_i,v_j) > d_{k-1}(v_i,v_k) + d_{k-1}(v_k,v_j)$, thì đặt

$$d_k(v_i,v_j) := d_{k-1}(v_i,v_k) + d_{k-1}(v_k,v_j)$$

và

$$p_k(v_i,v_j) := p_{k-1}(v_i,v_k)$$

ngược lại đặt

$$d_k(v_i,v_j) := d_{k-1}(v_i,v_j)$$

và

$$p_k(v_i,v_j) := p_{k-1}(v_i,v_j)$$

(3) Đặt $D = D_n$ là ma trận độ dài đường đi ngắn nhất, $P = P_n$.

♦ *Phương pháp xác định đường đi ngắn nhất từ đỉnh u đến đỉnh v*: Đường đi ngắn nhất từ *u* đến *v* gồm dãy các đỉnh

$$u, u_1, u_2, u_3, ..., u_k, u_{k+1}, ... , u_m, v$$

thỏa mãn

$$u_1 = p(u,v),\ u_2 = p(u_1,v),\ ...,\ u_{k+1} = p(u_k,v),\ ...,\ p(u_m,v) = v$$

◊ *Ghi chú*. Độ phức tạp của thuật toán Floyd-Warshall là $O(n^3)$.

• ***Định lý 2.3.6.*** Thuật toán Floyd-Warshall là đúng.

Chứng minh

Ở định lý 2.3.4 ta đã chứng minh D là ma trận độ dài đường đi ngắn nhất.

Bây giờ ta chỉ cần chứng minh thêm rằng phương pháp xây dựng đường đi theo ma trận P như ở thuật toán thực sự cho đường đi ngắn nhất.

Với mỗi cặp đỉnh (i,j) có $d_k(i,j) < +\infty$, ký hiệu

$$\mu_k(i,j) = (i,\ i_1,\ i_2,\ i_3,\ ...,\ i_h,\ i_{h+1},\ ...\ ,\ i_m,\ j)$$

là đường đi từ i đến j được xây dựng trên cơ sở ma trận P_k như sau:

$$i_1 = p_k(i,j),\ i_2 = p_k(i_1,j),\ ...,\ i_{h+1} = p_k(i_h,j),\ ...,\ p_k(i_m,j) = j$$

Ta chứng minh, bằng qui nạp theo k, bổ đề sau:

◊ *Bổ đề*. $\mu_k(i,j)$ có độ dài là $d_k(i,j)$, tức là đường đi ngắn nhất trong những đường nối đỉnh i đến j qua các đỉnh trung gian {1,2,...,k}.

- Bước cơ sở: Hiển nhiên mệnh đề đúng với k=0.

- Bước qui nạp: Giả sử mệnh đề đúng với k−1. Xét

$$\mu_k(i,j) = (i = i_0,\ i_1,\ i_2,\ i_3,\ ...,\ i_h,\ i_{h+1},\ ...\ ,\ i_m,\ i_{m+1} = j)$$

Ta xét hai khả năng sau:

(i) Trường hợp $d_k(i,j) = d_{k-1}(i,j)$: Suy ra

$p_k(i,j) = p_{k-1}(i,j) \Rightarrow\ i_1 = p_{k-1}(i,j) \in \{1,2,...,k-1\}$ và $(i,i_1) \in E$ (theo giả thiết quy nạp);

Ta chứng minh $d_k(i_1,j) = d_{k-1}(i_1,j)$

Giả sử ngược lại $d_k(i_1,j) < d_{k-1}(i_1,j)$. Khi đó ta có

$$d_{k-1}(i_1,j) > d_{k-1}(i_1,k) + d_{k-1}(k,j)$$

Suy ra

$$d_{k-1}(i,k) + d_{k-1}(k,j) \le d_{k-1}(i,i_1) + d_{k-1}(i_1,k) + d_{k-1}(k,j)$$

(bất đẳng thức tam giác)

$$< d_{k-1}(i,i_1) + d_{k-1}(i_1,j) \le d_{k-1}(i,j)$$

(bất đẳng thức tam giác)

Suy ra $d_k(i,j) < d_{k-1}(i,j)$, mâu thuẫn với $d_k(i,j) < d_{k-1}(i,j)$.

Vậy ta có $d_k(i_1,j) = d_{k-1}(i_1,j)$.

Từ đó, tương tự như trên, suy ra

$p_k(i_1,j) = p_{k-1}(i_1,j) \Rightarrow i_2 = p_{k-1}(i_1,j) \in \{1,2,...,k-1\}$ và $(i_1,i_2) \in E$.

Tương tự, với mọi $h=0, ..., m$, truy hồi ta nhận được

$p_k(i_h,j) = p_{k-1}(i_h,j)$ và $i_{h+1} = p_{k-1}(i_h,j) \in \{1,2,...,k-1\}$ và $(i_h,i_{h+1}) \in E$.

Vậy $\mu_k(i,j) \equiv \mu_{k-1}(i,j)$ là đường đi ngắn nhất trong những đường nối đỉnh i đến j qua các đỉnh trung gian $\{1,2,...,k-1\}$, có chiều dài bằng $d_{k-1}(i,j)$. Mà $d_k(i,j) = d_{k-1}(i,j)$, nên nó cũng đường đi ngắn nhất trong những đường nối đỉnh i đến j qua các đỉnh trung gian $\{1,2,...,k\}$, có chiều dài bằng $d_k(i,j)$.

(ii) Trường hợp $d_k(i,j) < d_{k-1}(i,j)$: Ta có

$$d_k(i,j) = d_{k-1}(i,k) + d_{k-1}(k,j) < d_{k-1}(i,j) \ \& \ p_k(i,j) = p_{k-1}(i,k)$$

Xét $i_1 = p_k(i,j) = p_{k-1}(i,k) \in \mu_{k-1}(i,k)$. Ta chứng minh

$$d_{k-1}(i_1,k) + d_{k-1}(k,j) < d_{k-1}(i_1,j) \qquad (*)$$

Giả sử ngược lại $d_{k-1}(i_1,k) + d_{k-1}(k,j) \geq d_{k-1}(i_1,j)$. Khi đó ta có

$d_{k-1}(i,i_1) + d_{k-1}(i_1,k) + d_{k-1}(k,j) \geq d_{k-1}(i,i_1) + d_{k-1}(i_1,j) \geq d_{k-1}(i,j)$

Mặt khác, do $i_1 = p_{k-1}(i,k) \in \{1,2,...,k-1\}$ nằm trên đường đi $\mu_{k-1}(i,k)$, nên $d_{k-1}(i,i_1) + d_{k-1}(i_1,k) = d_{k-1}(i,k)$. Suy ra $d_{k-1}(i,k) + d_{k-1}(k,j) \geq d_{k-1}(i,j)$, mâu thuẫn.

Như vậy (*) đúng, và ta có

$$d_k(i_1,j) = d_{k-1}(i_1,k) + d_{k-1}(k,j) \ \& \ p_k(i_1,j) = p_{k-1}(i_1,k)$$

kéo theo $i_2 = p_k(i_1,j) = p_{k-1}(i_1,k) \in \mu_{k-1}(i,k)$.

Truy hồi, ta chứng minh được rằng tồn tại h sao cho

$$\{i_3, i_4, ..., i_h = k\} \subset \mu_{k-1}(i,k).$$

Tức là

$$\mu_{k-1}(i,k) = \mu_k(i,k) \subset \mu_k(i,j)$$

Tiếp theo, ta có

$$d_k(i,k) + d_k(k,j) \geq d_k(i,j) = d_{k-1}(i,k) + d_{k-1}(k,j)$$

Mặt khác

$$d_k(i,k) \leq d_{k-1}(i,k) \text{ và } d_k(k,j) \leq d_{k-1}(k,j)$$

Suy ra

$$d_k(i,k) = d_{k-1}(i,k) \text{ và } d_k(k,j) = d_{k-1}(k,j)$$

Theo trường hợp (i) (áp dụng cho cặp đỉnh (k,j)), $\mu_k(k,j) = \mu_{k-1}(k,j)$ là đường đi ngắn nhất trong những đường nối đỉnh k đến j qua các đỉnh trung gian

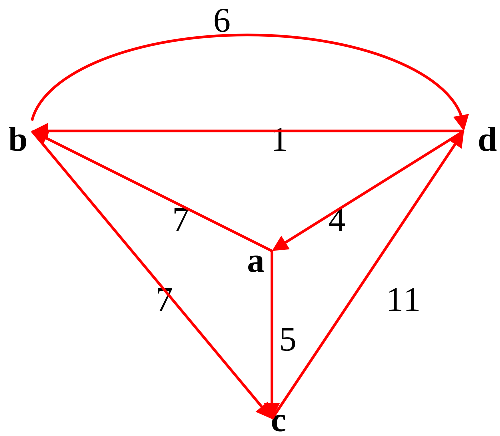
$\{1,2,...,k-1\}$, có chiều dài bằng $d_k(k,j)$. Mặt khác theo cách xây dựng $\mu_k(i,j)$, thì $\mu_k(i,j)$ là nối của hai đường đi $\mu_k(i,k)$ và $\mu_k(k,j)$: $\mu_k(i,j) = \mu_k(i,k) \cup \mu_k(k,j)$ và có độ dài bằng $d_k(i,j)$. Từ đó suy ra $\mu_k(i,j)$ là đường đi ngắn nhất trong những đường nối đỉnh i đến j qua các đỉnh trung gian $\{1,2,...,k\}$.

Bổ đề được chứng minh xong.

Từ bổ đề suy ra điều cần chứng minh vì $\mu(i,j) = \mu_n(i,j)$.

□

◊ *Ví dụ 2.3.3*. Xét đồ thị



Hình 2.3.10

Áp dụng giải thuật Floyd–Warshall ta nhận được các ma trận sau (quy ước: ở các ma trận D_k dấu trống là $+\infty$, ở các ma trận P_k dấu trống là không xác định):

- Các ma trận xuất phát:

$$D_0 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & 7 & 5 & \\ b & & & 7 & 6 \\ c & & & & 11 \\ d & 4 & 1 & & \end{array} \qquad P_0 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & b & c & \\ b & & & c & d \\ c & & & & d \\ d & a & b & & \end{array}$$

- Các ma trận cập nhật qua đỉnh **a**: (các giá trị mới được đóng khung)

$$D_1 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & 7 & 5 & \\ b & & & 7 & 6 \\ c & & & & 11 \\ d & 4 & 1 & \boxed{9} & \end{array} \qquad P_1 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & b & c & \\ b & & & c & d \\ c & & & & d \\ d & a & b & \boxed{a} & \end{array}$$

- Các ma trận cập nhật qua đỉnh **b**:

$$D_2 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & 7 & 5 & \boxed{13} \\ b & & & 7 & 6 \\ c & & & & 11 \\ d & 4 & 1 & \boxed{8} & \boxed{7} \end{array} \qquad P_2 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & b & c & \boxed{b} \\ b & & & c & d \\ c & & & & d \\ d & a & b & \boxed{b} & \boxed{b} \end{array}$$

- Các ma trận cập nhật qua đỉnh **c**:

$$D_3 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & 7 & 5 & 13 \\ b & & & 7 & 6 \\ c & & & & 11 \\ d & 4 & 1 & 8 & 7 \end{array} \qquad P_3 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & & b & c & b \\ b & & & c & d \\ c & & & & d \\ d & a & b & b & b \end{array}$$

(không có thay đổi).

- Các ma trận cập nhật qua đỉnh **d**:

$$D = D_4 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & \boxed{17} & 7 & 5 & 13 \\ b & \boxed{10} & \boxed{7} & 7 & 6 \\ c & \boxed{15} & \boxed{12} & \boxed{19} & 11 \\ d & 4 & 1 & 8 & 7 \end{array} \qquad P = P_4 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & \boxed{b} & b & c & b \\ b & \boxed{d} & \boxed{d} & c & d \\ c & \boxed{d} & \boxed{d} & \boxed{d} & d \\ d & a & b & b & b \end{array}$$

Cuối cùng, ta có ma trận khoảng cách ngắn nhất giữa các đỉnh $D = D_4$. Ta thấy đồ thị liên thông mạnh và chứa chu trình.

Sử dụng ma trận $P = P_4$, ta có thể tìm đường đi ngắn nhất giữa các đỉnh. Chẳng hạn, để tìm đường đi từ đỉnh *d* đến đỉnh *c* ta làm như sau:

Đặt

$$i_1 := P(d,c) = b;\ i_2 := P(b,c) = c$$

Từ đó ta nhận được đường đi ngắn nhất từ d đến c:

$$d \to b \to c$$

với độ dài là 8.

6. Ứng dụng

6.1. Bài toán chọn địa điểm

Bài toán tìm địa điểm đặt cơ sở dịch vụ trong cộng đồng (trường học, bưu điện, phòng cháy chữa cháy, ...) sao cho kinh tế và hiệu quả nhất có nhiều ứng dụng trong thực tế. Bài toán này thường được mô hình hoá như mạng lưới dạng đồ thị, trong đó các cơ sở có thể được bố trí ở một hoặc vài vị trí.

Nếu các cơ sở là các cơ quan như trường học, hay bưu điện, thì nên đặt chúng ở các vị trí sao cho tổng khoảng cách các vùng khác nhau đến cơ sở là nhỏ nhất. Lớp bài toán này gọi là ***Bài toán cực tiểu tổng*** **(*minsum problem*)**.

Mặt khác, nếu các cơ sở là các cơ quan như phòng cháy chữa cháy, thì nên đặt chúng ở các vị trí sao cho khoảng cách từ cơ sở đến điểm xa nhất của cộng đồng là nhỏ nhất. Lớp bài toán này gọi là ***Bài toán cực tiểu trị lớn nhất*** **(*minmax problem*)**.

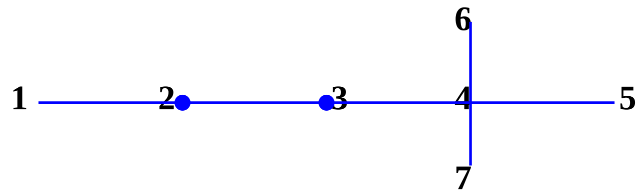
• *Bài toán cực tiểu tổng*

Cho trọng đồ $G = (V,E,c)$, trong đó $V = \{1, 2, ..., n\}$, với trọng số cạnh không âm. Cho $D = [d(i,j)]$ là ma trận khoảng cách nhỏ nhất của G (bằng thuật toán Floyd-Warshal). Với mỗi đỉnh $i \in V$, đặt $s(i)$ là tổng tất cả phần tử trên hàng i của ma trận D, trừ phần tử $d(i,i)$. Đỉnh j gọi là *đỉnh cực tiểu tổng* nếu $s(j) \leq s(i)\ \forall i \in V$.

• *Bài toán cực tiểu trị lớn nhất* (*Bài toán tìm tâm đồ thị*)

Cho trọng đồ $G = (V,E,c)$, trong đó $V = \{1, 2, ..., n\}$, với trọng số cạnh không âm. Cho $D = [d(i,j)]$ là ma trận khoảng cách nhỏ nhất của G (bằng thuật toán Floyd-Warshal). Với mỗi đỉnh $i \in V$, đặt e(i) là phần tử lớn nhất (độ lệch tâm) trên hàng i của ma trận D, trừ phần tử d(i,i). Đỉnh j gọi là *đỉnh tâm* nếu e(j) $\le$ e(i) $\forall i \in V$. Tập tất cả các đỉnh tâm gọi là *tâm đồ thị*.

◊ *Ví dụ*. Tìm các đỉnh cực tiểu tổng, cực tiểu trị lớn nhất của đồ thị sau



với trọng số tất cả các cạnh là 1.

Giải

Ma trận khoảng cách ngắn nhất của đồ thị là

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 4 & 4 \\ 1 & 0 & 1 & 2 & 3 & 3 & 3 \\ 2 & 1 & 0 & 1 & 2 & 2 & 2 \\ 3 & 2 & 1 & 0 & 1 & 1 & 1 \\ 4 & 3 & 2 & 1 & 0 & 2 & 2 \\ 4 & 3 & 2 & 1 & 2 & 0 & 2 \\ 4 & 3 & 2 & 1 & 2 & 2 & 0 \end{bmatrix}$$

Tổng các hàng tương ứng là 18, 13, 10, 9, 14, 14, 14. Như vậy đỉnh 4 là đỉnh cực tiểu tổng duy nhất và tập đỉnh cực tiểu tổng là {4}.

Độ lệch tâm của các đỉnh tương ứng là 4, 3, 2, 3, 4, 4, 4. Như vậy đỉnh 3 là đỉnh cực tiểu độ lệch tâm duy nhất và tâm đồ thị là {3}.

6.2. Bài toán người đưa thư

Nhân viên bưu điện nhận thư ở bưu cục, sau đó đi qua tất cả đường phố thuộc khối phố của mình phát thư rồi quay lại bưu cục. Tìm lộ trình ngắn nhất cho người đưa thư.

Giả sử G=(V,E,w) là đồ thị biểu diễn bản đồ đường phố. Ký hiệu U là tập tất cả đỉnh bậc lẻ, |U| = 2*k*. Ký hiệu S là tập các phân hoạch *k* cặp phần tử của U. Gọi s(k) là lực lượng của S.

Ta lập công thức truy hồi tính s(k). Cố định $u \in U$ bất kỳ. Với mỗi $v \in U\backslash\{u\}$, cặp (u,v) cùng với mỗi phân hoạch (k-1) cặp của U\{u,v} tạo thành một phân hoạch k cặp của U. Suy ra s(k) = (2k-1).s(k-1). Thế liên tiếp ta có

$$s(k) = (2k-1).(2k-3)\ldots..1 = (2k-1)!!.$$

Thuật toán tìm lộ trình đưa thư ngắn nhất như sau.

Bước 1. *Khởi tạo*. Tính tập đỉnh bậc lẻ U, |U| = 2*k*. Tính khoảng cách d(u,v) từng cặp phần tử (u,v) của U (sử dụng thuật toán Floyd-Warshall hoặc Dijkstra). Đặt *min* = ∞.

Bước 2. Tìm phân hoạch s_{min} có tổng khoảng cách nhỏ nhất.

Với mỗi phân hoạch s gồm k cặp phần tử của U thực hiện

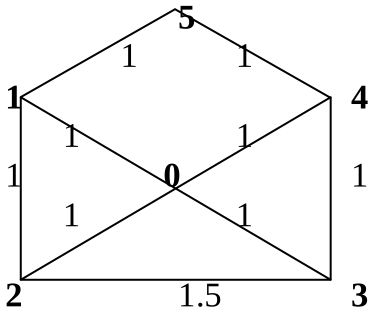
{Tính tổng T(s) = $\sum_{(u,v)\in s} d(u,v)$

Nếu T(s) < min, thì đặt min = T(s) và s_{min} := s;}

Bước 3. Lập đồ thị G' = (V, E'), trong đó E' = $E \cup s_{min}$. Bậc các đỉnh trong G' có bậc chẵn, suy ra đồ thị G' có chu trình Euler. Tìm chu trình Euler C' của G'.

Bước 4. Thay mỗi cạnh (u,v) $\in s_{min}$ bằng đường đi ngắn nhất trong G, ta nhận được lộ trình đưa thư ngắn nhất.

◊ *Ví dụ*. Xét bài toán người đưa thư cho ở đồ thị G có các đỉnh 0,1,2,3,4,5 và các cạnh với trọng số cho ở hình sau



G

Tập đỉnh bậc lẻ U = {1,2,3,4} có 3 phân hoạch

s_1 = {[1,2],[3,4]}, s_2 = {[1,3],[2,4]} và s_3 = {[1,4],[2,3]}.

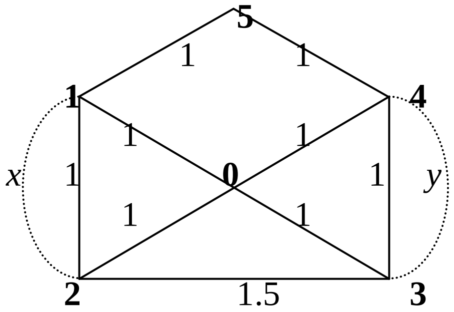
Ta có

$T(s_1) = d(1,2)+d(3,4) = 1+1 = 2$,

$T(s_2) = d(1,3)+d(2,4) = 2+2 = 4$,

$T(s_3) = d(1,4)+d(2,3) = 2+1.5 = 3.5$.

Vậy $s_{min} = s_1$. Thêm hai cạnh ảo *x* = (1,2) và *y* = (3,4) với độ dài 1 ta nhận được đồ thị G' sau



G'

Sử dụng thuật toán tìm chu trình Euler ta dễ dàng tìm được chu trình Euler của đồ thị G' sau

0→1→2→3→4→5→1-*x*→2→0→3-*y*→4→0

Thay các cạnh ảo bằng đường đi thật ta nhận được lộ trình đưa thư ngắn nhất trong đồ thị G như sau

0→1→2→3→4→5→1→2→0→3→4→0

với tổng độ dài là

1+1+1.5+1+1+1+1+1+1+1+1 = 11.5

◊ *Ghi chú*. Ở bước 2 có thể sử dụng thuật toán quay lui để liệt kê tất cả các phân hoạch *k* cặp phần tử của U.

Giả sử U={1,2,…,2k} và phân hoạch k cặp phần tử của U được biểu diễn bằng dãy

(x[1], y[1]), …, (x[r], y[r]), …, (x[k], y[k])

thỏa x[1] < x[2] < ... < x[k] và x[r] < y[r], ∀r=1,…,k.

Ký hiệu status[i] là trạng thái của đỉnh i∈U: status[i]=true nghĩa là i được ghép cặp, ngược lại i chưa được ghép cặp. Khởi tạo status[i]:=false, ∀i=1,…,2k. Thủ tục quay lui đệ quy như sau:

```
Procedure Try(r:integer);
var i, j:integer;
begin
    for i:=1 to 2k  do
     for j:=i+1 to 2k do
     if (i > x[r-1]) and not status[i] and not status[j]  then
         begin
          x[r] := i; y[r]:=j;
          status[i]:= true; status[j]:= true;
          if  r = k  then  <xử lý phân hoạch ở bước 2 trong thuật toán>
          else  Try(r+1);
          status[i]:= false; status[j]:= false;
         end;
end;
```

Trong chương trình chính gán x[0] := 0 và gọi: *Try*(1).

IV. ĐƯỜNG ĐI HAMILTON

1. Đường đi, chu trình Hamilton

• ***Định nghĩa 2.4.1***

Cho đồ thị (có hướng) G=(V,E).

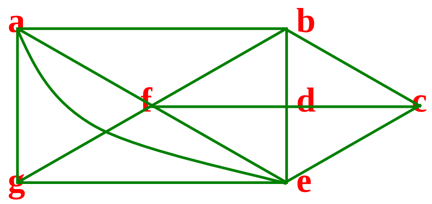
Chu trình (có hướng) Hamilton là chu trình (có hướng) sơ cấp qua mọi đỉnh đồ thị.

Đường đi (có hướng) Hamilton là đường đi (có hướng) sơ cấp qua mọi đỉnh đồ thị.

Như vậy mọi chu trình Hamilton có độ dài bằng số đỉnh, và mọi đường đi Hamilton có độ dài bằng số đỉnh trừ 1.

Đồ thị chứa chu trình (có hướng) Hamilton gọi là *đồ thị Hamilton.*

◊ *Ví dụ 2.4.1.* Xét đồ thị



Hình 2.4.1

Đồ thị trên không có đường đi Euler vì các đỉnh g, c, e, f có bậc lẻ, nhưng có chu trình Hamilton

$$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow a$$

2. Điều kiện cần

• ***Định lí 2.4.1***

Giả sử đồ thị G có chu trình Hamilton C. Khi đó

(i) Đồ thị G liên thông.

(ii) Mọi đỉnh của G có bậc lớn hơn hoặc bằng 2, và có đúng hai cạnh liên thuộc nằm trên chu trình C.

(iii) Nếu xóa đi *k* đỉnh bất kỳ cùng các cạnh liên thuộc chúng, thì đồ thị còn lại sẽ có tối đa *k* thành phần liên thông.

Chứng minh

Mệnh đề (i) và (ii) là hiển nhiên. Mệnh đề (iii) suy ra từ thực tế là khi xóa đi k đỉnh bất kỳ cùng các cạnh liên thuộc chúng thì chu trình C bị chia ra thành nhiều nhất k phần.

□

• ***Hệ quả 2.4.2***

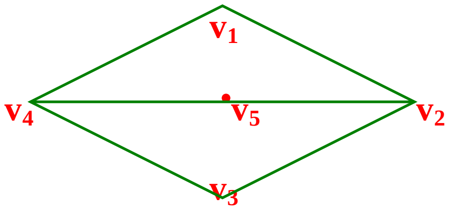
Giả sử đồ thị *n* đỉnh G có đường đi Hamilton P. Khi đó

(i) Đồ thị G liên thông.

(ii) Có ít nhất $n-2$ đỉnh bậc ≥ 2, và mỗi đỉnh đó có đúng hai cạnh liên thuộc nằm trên đường đi P.

(iii) Nếu xóa đi *k* đỉnh bất kỳ cùng các cạnh liên thuộc chúng, thì đồ thị còn lại sẽ có tối đa $k+1$ thành phần liên thông.

◊ *Ví dụ 2.4.2.* Xét đồ thị



Hình 2.4.2

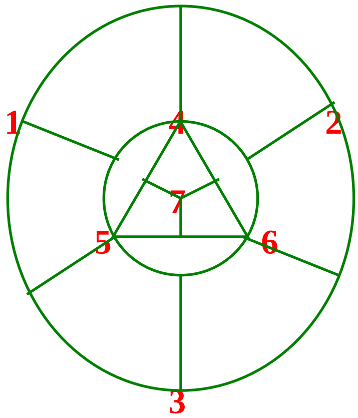
Đồ thị có đường đi Hamilton, ví dụ: v_1-v_2-v_3-v_4-v_5.

Đồ thị không có chu trình Hamilton.

Thật vậy, nếu tồn tại chu trình Hamilton C thì nó phải có 5 cạnh. Vì bậc $deg(v_2) = deg(v_4) = 3$ nên phải một cạnh tới v_2 và một cạnh tới v_4 không thuộc chu trình C. Số cạnh còn lại là 4 nên C không thể có 5 cạnh được, mâu thuẫn.

Ta cũng có thể áp dụng trực tiếp Định lý 2.4.1. Nếu bỏ đi 2 đỉnh v_2 và v_4 cùng các cạnh liên thuộc chúng thì đồ thị còn lại là 3 đỉnh độc lập, có 3 thành phần liên thông. Như vậy theo mệnh đề (iii) của Định lý 2.4.1 thì đồ thị không có chu trình Hamilton.

◊ *Ví dụ 2.4.3*. Chứng minh rằng đồ thị sau không có đường đi Hamilton.



Hình 2.4.3

Giả sử P là đường đi Hamilton. Đồ thị trên có 16 đỉnh, nên đường đi P có 15 cạnh (*).

- *Cách 1*. Vì đồ thị có 27 cạnh nên số cạnh không nằm trên đường P phải là 12 (**).

Xét các đỉnh thuộc tập U={1,2,3,4,5,6,7}. Gọi E_i , i=1,...,7, là tập hợp các cạnh liên thuộc đỉnh $i \in U$ không nằm trên đường đi P. Vì mọi cặp đỉnh trong U không kề nhau nên các tập E_i cũng rời nhau từng cặp một. Như vậy số cạnh liên thuộc các đỉnh của U không nằm trên P là

$$k = \sum_{i=1}^{7} card(E_i)$$

Vì số cạnh liên thuộc 1 đỉnh nằm trên đường đi P nhiều nhất là 2 nên mỗi tập E_1, E_2, E_3, E_7 có ít nhất 1 cạnh và mỗi tập E_4, E_5, E_6 có ít nhất 3 cạnh. Vì vậy

$$k \geq 4*1 + 3*3 = 13.$$

mâu thuẫn với (**).

- *Cách 2*. Xét 9 đỉnh còn lại. Đây là các đỉnh không kề nhau từng cặp một, trong đó có ít nhất 7 đỉnh có 2 cạnh liên thuộc $\in$ P và 2 đỉnh còn lại có ít nhất 1 cạnh liên thuộc ∈ P. Như vậy số cạnh ∈ P ít nhất là 7*2+2 = 16, mâu thuẫn với (*).

- *Cách 3*. Xoá 7 đỉnh 1, 2, 3, 4, 5, 6, 7 và các cạnh liên thuộc thì đồ thị còn lại có 9 đỉnh cô lập, tức 9 thành phần liên thông. Theo hệ quả 2.4.2 (iii), đồ thị không có đường đi Hamilton.

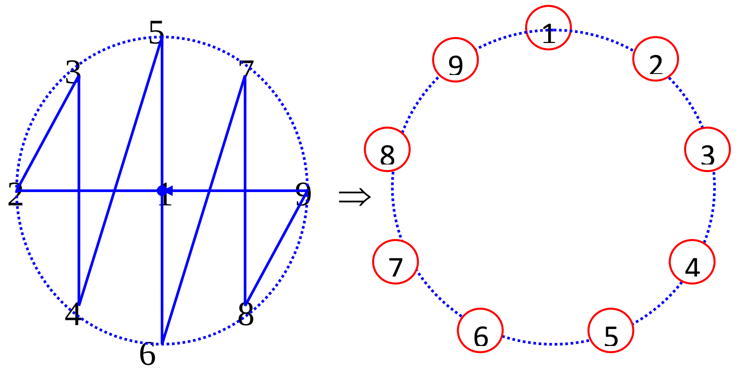
◊ *Ví dụ 2.4.4*. (*Bài toán xếp chỗ ngồi*) 9 người bạn cùng ngồi ăn trong bàn tròn 4 lần. Mỗi lần họ được xếp ngồi theo một thứ tự. Hãy thay đổi chỗ ngồi mỗi lần sao cho không có 2 người ngồi gần nhau hơn 1 lần.

Giải

Ta lập đồ thị 9 đỉnh 1, 2, ...,9, đỉnh *i* chỉ người *i*. Ta đặt đỉnh 1 tại tâm và các đỉnh còn lại trên đường tròn như hình vẽ. Mỗi cách xếp là một chu trình Hamilton của đồ thị.

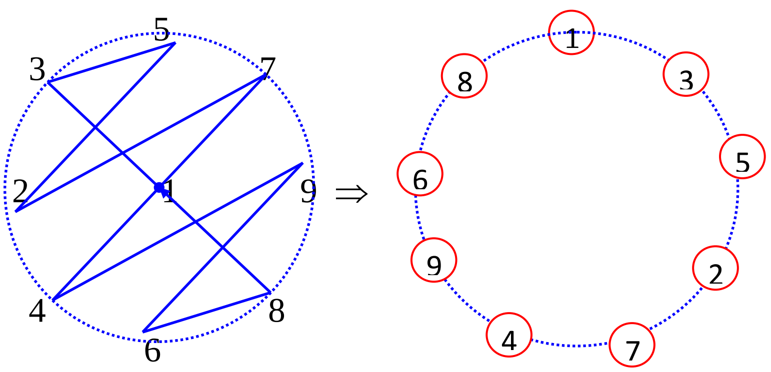
Chu trình thứ nhất như hình vẽ là

$$1\to2\to3\to4\to5\to6\to7\to8\to9\to1$$



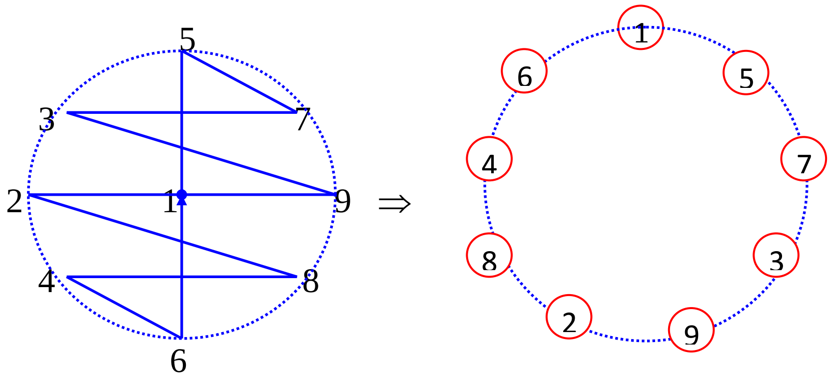
Hình 2.4.4

Xoay lần lượt chu trình các góc π/4 theo chiều kim đồng hồ ta nhận được các chu trình, cũng là các cách xếp, sau:



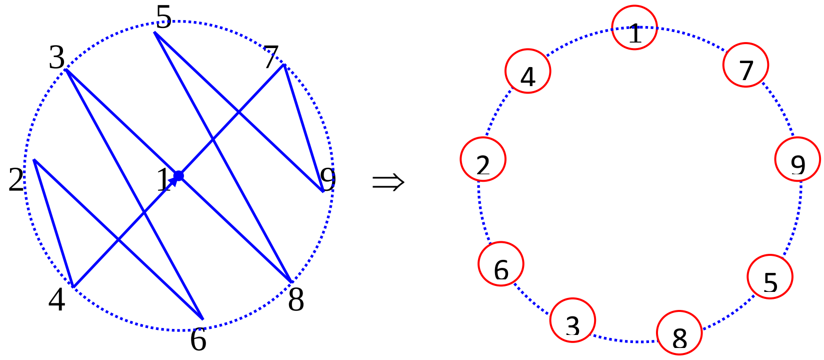
Hình 2.4.5

$$1\to3\to5\to2\to7\to4\to9\to6\to8\to1$$



Hình 2.4.6

1→5→7→3→9→2→8→4→6→1



Hình 2.4.7

1→7→9→5→8→3→6→2→4→1

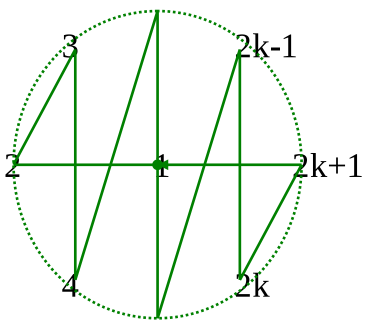
3. Điều kiện đủ

• ***Định lý 2.4.3.*** Đồ thị đủ K_n với n lẻ ($n \geq 3$) có $(n-1)/2$ chu trình Hamilton từng đôi một không giao nhau (tức là không có cạnh chung).

Chứng minh

Tương tự như lời giải bài toán xếp 9 người trên bàn tròn, ta xây dựng cách xếp theo chu trình Hamilton trên đồ thị sau (n=2k+1):

1→2→3→ ... →2k→2k+1→1



Hình 2.4.8

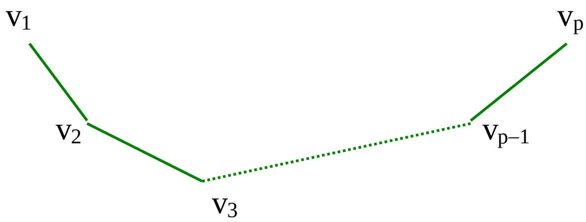
Xoay chu trình lần lượt một góc π/k theo chiều kim đồng hồ ta nhận được k chu trình.

• ***Định lý 2.4.4*** (*Dirac*)**.** Cho G là đơn đồ thị n đỉnh (n≥3). Nếu bậc $deg(v) \geq n/2$ với mọi đỉnh v của G, thì G có chu trình Hamilton.

Chứng minh

(i) Hiển nhiên định lý đúng với $n = 3$.

(ii) Giả sử $n > 3$. Gọi $P=(v_1,...,v_p)$ là đường đi sơ cấp dài nhất trong G, có độ dài là $p-1$. Vậy mọi đỉnh kề với v_1 và mọi đỉnh kề với v_p phải thuộc P.

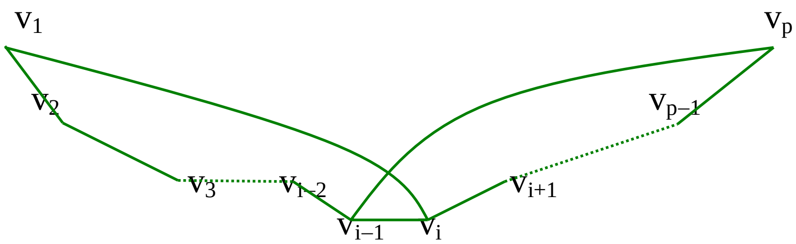


Hình 2.4.9

Nếu $p-1 < n/2$ thì đỉnh v_p phải kề với đỉnh v_{p+1} khác $v_1,...,v_{p-1}$ (vì $deg(v_p) \geq n/2$), điều này mâu thuẫn với giả thiết P là đường dài nhất. Vậy phải có

$$p - 1 \geq n/2 \Leftrightarrow p \geq 1 + n/2$$

Tiếp theo ta chỉ ra rằng phải tồn tại đỉnh v_i với $2 \leq i \leq p$ sao cho v_1 kề với v_i và v_p kề với v_{i-1} (xem hình).



Hình 2.4.10

Thật vậy, nếu không tồn tại một v_i như vậy, thì do v_1 kề với ít nhất $n/2$ đỉnh trên P, nên v_p sẽ không kề với ít nhất $n/2$ đỉnh trên P. Suy ra bậc của v_p thoả

$$deg(v_p) \leq (n-1) - n/2 < n/2$$

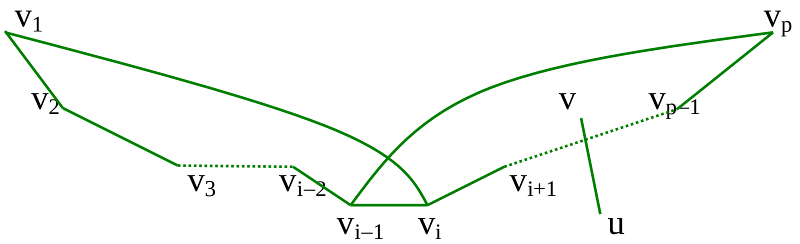
Điều này mâu thuẫn với giả thiết $deg(v) \geq n/2$ với mọi đỉnh v.

Như vậy G có chu trình

$$C = (v_1, v_i, v_{i+1},..., v_{p-1}, v_p, v_{i-1}, v_{i-2}, ..., v_2, v_1)$$

chứa mọi đỉnh của P.

Nếu có đỉnh u không thuộc C thì vì $p \geq 1 + n/2$ nên số đỉnh không thuộc C nhiều nhất là $n/2 - 1$. Từ giả thiết $deg(u) \geq n/2$ suy ra u phải kề đỉnh v nào đó thuộc C (xem hình 2.4.11).



Hình 2.4.11

Mặt khác cạnh (u,v) thêm vào chu trình C tạo thành đường đi dài lớn hơn P, mâu thuẫn với giả thiết P là đường đi dài nhất. Vậy C chứa mọi đỉnh của G, tức C là chu trình Hamilton.

□

• ***Định lý 2.4.5.*** Cho G là đồ thị đơn *n* đỉnh ($n \geq 3$). Nếu bậc $deg(v) \geq (n-1)/2$ với mọi đỉnh *v* của G, thì G có đường đi Hamilton.

Chứng minh

Nếu n = 1 thì G có đường Hamilton tầm thường là 1 đỉnh.

Giả sử n > 1. Ta lập đồ thị H bằng cách thêm vào G đỉnh v và tất cả các cạnh nối v với mọi đỉnh của G. Đồ thị H có n + 1 đỉnh và deg(v) = n. Với mọi đỉnh $u \in G$ ta có

$$deg_H(u) = deg_G(u) + 1 \geq (n-1)/2 + 1 = (n+1)/2$$

Theo định lý Dirac thì H có chu trình Hamilton C. Bỏ đi v và các cạnh tới v ta được đường đi Hamilton trong G.

□

• ***Định lý 2.4.6.*** Cho G là đồ thị đơn *n* đỉnh ($n \geq 3$). Giả sử u và v là hai đỉnh không kề nhau của G sao cho

$$deg(u) + deg(v) \geq n.$$

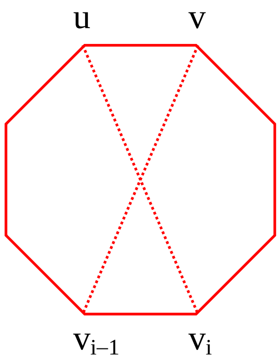
Khi đó G có chu trình Hamilton khi và chỉ khi đồ thị G+(u,v) (đồ thị G thêm cạnh (u,v)) có chu trình Hamilton.

Chứng minh

Nếu G có chu trình Hamilton C thì đó cũng là chu trình của G+(u,v).

Giả sử C là chu trình Hamilton của G+(u,v). Nếu $(u,v) \notin C$ thì C cũng là chu trình Hamilton của G. Giả sử $(u,v) \in C$. Khi đó chu trình Hamilton C có thể biểu diễn như sau

$$C = (u = v_1, v_2, \ldots, v_{n-1}, v_n = v, u)$$



Hình 2.4.12

Tồn tại đỉnh v_i, $3 \le i \le n-1$, sao cho v_1 kề v_i và v_n kề v_{i-1} . Thật vậy, giả sử không có đỉnh v_i như vậy. Gọi k là số đỉnh kề u trong G. Số đỉnh không kề v trong G sẽ lớn hơn hoặc bằng k, như vậy số đỉnh kề v trong G không quá n−1−k. Suy ra

$$\deg(u) + \deg(v) \le k + (n-1-k) = n-1$$

mâu thuẫn với giả thiết định lý.

Chu trình

$$(v_1, v_i, v_{i+1},..., v_{p-1}, v_p, v_{i-1}, v_{i-2}, ..., v_2, v_1)$$

sẽ là chu trình Hamilton trong G.

□

• ***Định lý 2.4.7.*** Cho G là đồ thị đơn giản n đỉnh. Giả sử G' và G" là hai đồ thị thu được từ G bằng quy nạp nối tất cả cặp đỉnh không kề nhau có tổng các bậc ít nhất bằng n. Khi đó ta có G' = G".

Chứng minh

Giả sử $e_1,..., e_m$ và $f_1,...,f_n$ là các cạnh thêm vào G để có tương ứng G' và G". Ta chứng minh mỗi cạnh e_i phải thuộc G" và mỗi cạnh f_j phải thuộc G'.

Giả sử có cạnh e_i không thuộc G". Gọi k là chỉ số nhỏ nhất sao cho e_{k+1}=(u,v) không thuộc G". Đặt $H = G + \{e_1,..., e_k\}$. H là đồ thị con của G' và G". Ta có

$$\deg_H(u) + \deg_H(v) \ge n$$

Do đó

$$\deg_{G''}(u) + \deg_{G''}(v) \ge \deg_H(u) + \deg_H(v) \ge n$$

Điều này vô lý vì u, v không kề nhau trong G". Như vậy mỗi cạnh e_i phải thuộc G". Tương tự mỗi f_j phải thuộc G', do đó G' = G".

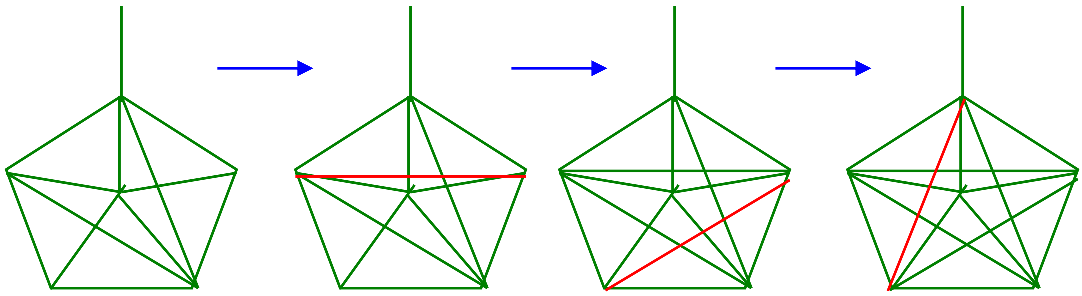
□

Từ định lý trên ta có thể định nghĩa khái niệm *bao đóng* của đồ thị.

• ***Định nghĩa 2.4.2***

Bao đóng C(G) của đồ thị G *n* đỉnh là đồ thị thu được từ G bằng cách, theo quy nạp, nối tất cả các cặp đỉnh không kề nhau mà tổng số bậc ít nhất bằng *n* cho đến khi không còn cặp đỉnh nào như vậy nữa.

◊ *Ví dụ 2.4.5*. Minh hoạ cách xây dựng bao đóng.



Hình 2.4.13

• ***Định lý 2.4.8.*** Đồ thị G có chu trình Hamilton khi và chỉ khi bao đóng của G có chu trình Hamilton.

Chứng minh. Suy ra từ định lý 2.4.6.

• ***Định lý 2.4.9.*** Nếu bao đóng $C(G) = K_n$ $(n \geq 3)$ thì đồ thị G có chu trình Hamilton.

Chứng minh. Vì đồ thị K_n có chu trình Hamilton nên G có chu trình Hamilton theo Định lý 2.4.8.

• ***Định lý 2.4.10*** (*Định lý Ore*)**.** Cho G là đơn đồ thị n đỉnh $(n \geq 3)$. Nếu $\deg(u)+\deg(v) \geq n$ với mọi cặp đỉnh không kề nhau thì đồ thị G có chu trình Hamilton.

Chứng minh

Vì bao đóng C(G) là đồ thị đủ K_n nên C(G) có chu trình Hamilton, suy ra G có chu trình Hamilton theo Định lý 2.4.9.

• ***Định lý 2.4.11.*** Cho G là đơn đồ thị n đỉnh $(n \geq 3)$ và m cạnh. Nếu $m \geq C(n-1,2)+2$ thì đồ thị G có chu trình Hamilton.

Chứng minh

Nếu G là đồ thị đủ thì G có chu trình Hamilton. Giả sử G không là đồ thị đủ và thoả giả thiết định lý. Gọi u, v là hai đỉnh không kề nhau trong G. Gọi H là đồ thị thu được từ G bằng cách bỏ đi hai đỉnh u, v và các cạnh liên thuộc với chúng. Khi đó số cạnh của G bằng số cạnh của H cộng với deg(u)+deg(v). Hơn nữa vì số cạnh của H không quá $C(n-2,2)$ nên

$$\deg(u)+\deg(v) = (\text{số cạnh của G}) - (\text{số cạnh của H}) \geq C(n-1,2)+ 2 - C(n-2,2) = n$$

Suy ra G có chu trình Hamilton.

□

• ***Định lý 2.4.12.*** Cho đồ thị đơn G là đồ thị lưỡng phân với hai tập đỉnh V_1 và V_2 sao cho

$$\text{card}(V_1) = \text{card}(V_2) = n \geq 2$$

Nếu bậc $\deg(v) > n/2$ với mọi đỉnh v của G, thì G có chu trình Hamilton.

Chứng minh

Ta chứng minh bằng phản chứng. Giả sử G là đồ thị có số cạnh tối đa thoả điều kiện định lý nhưng không có chu trình Hamilton. Vì $K_{n,n}$ có chu trình Hamilton với mọi $n \geq 2$, nên G phải có hai đỉnh không kề nhau $u \in V_1$, $v \in V_2$. Đồ thị G+(u,v) có chu trình Hamilton (vì G là đồ thị không có chu trình Hamilton có số cạnh tối đa), và chu trình này phải chứa (u,v). Do đó G có đường đi Hamilton từ u đến v. Ký hiệu đường đó là

$$P = (u = v_1, v_2, \ldots, v_{2n-1}, v_{2n} = v)$$

với $v_i \in V_1$ nếu i lẻ và $v_i \in V_2$ nếu i chẵn.

Tiếp theo, nếu (v_1, v_i) là cạnh của G thì (v_{i-1}, v_{2n}) không phải là cạnh của G, vì ngược lại ta có thể xây dựng chu trình Hamilton của G giống chứng minh định lý Dirac.

Vì $\deg(v_1) > n/2$, suy ra

$$\deg(v_{2n}) < n - n/2 = n/2,$$

và điều này mâu thuẫn với giả thiết định lý.

□

4. Đồ thị có hướng.

Cho đồ thị có hướng G với *n* đỉnh. Ta có các kết quả phát biểu trong các định lý sau (xem [10]).

• ***Định lý 2.4.13.*** (*điều kiện đủ tồn tại chu trình có hướng Hamilton*)

a. (Meyniel) Nếu đồ thị G liên thông mạnh và

$$\deg(u) + \deg(v) \geq 2n-1 \quad \forall u, v \in G \text{ không kề nhau}$$

thì G có chu trình có hướng Hamilton.

b. (Ghoula-Houri) Nếu đồ thị G liên thông mạnh và

$$\deg(v) \geq n \quad \forall v \in G$$

thì G có chu trình có hướng Hamilton.

c. (Woodall) Nếu

$$\deg_O(u) + \deg_I(v) \geq n \quad \forall u, v \in G \text{ không tồn tại cung từ u đến v}$$

thì G có chu trình có hướng Hamilton.

d. Nếu

$$\deg_I(v) \geq n/2 \;\; \& \;\; \deg_O(v) \geq n/2 \quad \forall v \in G$$

thì G có chu trình có hướng Hamilton.

• ***Định lý 2.4.14.*** (*điều kiện đủ tồn tại đường đi có hướng Hamilton*)

a. Nếu

$$\deg(u) + \deg(v) \geq 2n-3 \quad \forall u, v \in G \text{ không kề nhau}$$

thì G có đường đi có hướng Hamilton.

b. Nếu

$$\deg(v) \geq n - 1 \quad \forall v \in G$$

thì G có đường đi có hướng Hamilton.

c. Nếu

$$\deg_O(u) + \deg_I(v) \geq n - 1 \quad \forall u, v \in G \text{ không tồn tại cung từ u đến v}$$

thì G có đường đi có hướng Hamilton.

d. Nếu

$$\deg_I(v) \geq n/2 \ \& \ \deg_O(v) \geq n/2 \ \ \forall v \in G$$

thì G có đường đi có hướng Hamilton.

Bây giờ chúng ta sẽ nghiên cứu đường đi và chu trình có hướng Hamilton trong đồ thị có hướng đủ (có đồ thị lót đủ). Trước hết là định lý khẳng định sự tồn tại đường đi có hướng Hamilton trong đồ thị có hướng đủ.

• ***Định lý 2.4.15 (Konig).*** Mọi đồ thị có hướng đủ đều có đường đi có hướng Hamilton.

Chứng minh

Xét đồ thị có hướng đầy đủ n đỉnh G = (V,E). Trong các đường đi sơ cấp của G, tồn tại đường đi dài nhất. Ký hiệu

$$p = (v_1 \to v_2 \to \ldots \to v_k)$$

là một đường đi sơ cấp dài nhất. Bằng phản chứng ta chứng minh mọi đỉnh của G đều thuộc p, và như vậy p là đường đi Hamilton.

Giả sử tồn tại đỉnh u không thuộc đường đi p. Khi đó sẽ tồn tại đường đi p' thuộc một trong ba dạng sau:

(i) $p' = (u \to v_1 \to v_2 \to \ldots \to v_k)$ (tức cung $(u \to v_1) \in E$)

(ii) $\exists$ i, $1 \leq i < k$, $p' = (v_1 \to \ldots v_i \to u \to v_{i+1} \ldots \to v_k)$ (tức các cung $(v_i \to u)$, $(u \to v_{i+1}) \in E$)

(iii) $p' = (v_1 \to v_2 \to \ldots \to v_k \to u)$ (tức cung $(v_k \to u) \in E$)

Thật vậy, nếu không tồn tại đường đi dạng (i) và (ii) thì do tính đầy đủ của đồ thị ta có

$$(v_1 \to u) \in E \Rightarrow (v_2 \to u) \in E \Rightarrow \ldots \Rightarrow (v_{k-1} \to u) \in E \Rightarrow (v_k \to u) \in E$$

tức là phải tồn tại đường đi dạng (iii).

Đường đi sơ cấp p' dài hơn đường đi p, mâu thuẫn với giả thiết p là đường đi dài nhất.

□

• ***Định nghĩa 2.4.2.*** Đồ thị có hướng đủ G = (V, E) gọi là *bắc cầu* nếu (u,v), (v,w) $\in$ E suy ra (u,w) $\in$ E.

Từ định nghĩa ta thấy ngay, một *đồ thị có hướng đủ là bắc cầu khi và chỉ khi nó không có chu trình có hướng độ dài 3.*

• ***Định lý 2.4.16.*** Đồ thị có hướng đủ bắc cầu khi và chỉ khi nó không có chu trình có hướng.

Chứng minh

Điều kiện đủ là hiển nhiên.

Ngược lại cho G=(V,E) là đồ thị có hướng đủ bắc cầu. Giả sử G có chu trình có hướng độ dài k > 3

$$v_1 \to v_2 \to v_3 \to \dots v_{k-1} \to v_k \to v_1$$

Do tính bắc cầu nên ta có cung $(v_{k-1}, v_1) \in E$. Suy ra G có chu trình có hướng độ dài (k−1). Lập luận tương tự suy ra G có chu trình có hướng độ dài k−2, k−3, ... cho đến khi G có chu trình có hướng độ dài 3, mâu thuẫn với tính bắc cầu.

□

• ***Định lý 2.4.17.*** Đường đi Hamilton trong đồ thị có hướng đủ là duy nhất khi và chỉ khi đồ thị bắc cầu.

Chứng minh

(i) Cho G là đồ thị có hướng đủ bắc cầu. Giả sử có 2 đường đi Hamilton khác nhau P_1 và P_2. Khi đó sẽ có 2 đỉnh u, v thoả u đứng trước v trong P_1 và v đứng trước u trong P_2. Suy ra tồn tại chu trình $P_1(u,v) \cup P_2(v,u)$, mâu thuẫn với tính bắc cầu.

(ii) Cho G là đồ thị có hướng đủ có duy nhất đường đi có hướng Hamilton

$$P = v_1 \to v_2 \to v_3 \to \dots v_{n-1} \to v_n$$

Nếu với mọi $1 \le i < j \le n$, ta có cung $(v_i, v_j) \in E$ (tức không có cung ngược hướng), thì với mọi $1 \le i < j < k \le n$, các đỉnh v_i, v_j và v_k không tạo thành chu trình có hướng được, suy ra G là đồ thị bắc cầu.

Ngược lại giả sử tồn tại $1 \le i < j \le n$ với cung $(v_j, v_i) \in E$. Ta sẽ chỉ ra sự tồn tại đường đi có hướng Hamilton khác P. Ta chọn

$$a := \min\{ i \mid 1 \le i \le n \ \&\ \exists\, i < j \le n : (v_j, v_i) \in E \}$$

và

$$b := \max\{ j \mid a < j \le n : (v_j, v_a) \in E \}$$

Ta xét các trường hợp sau.

- Trường hợp $1 < a < b < n$: Xét các đỉnh v_{a-1} và v_{a+1}. Theo cách chọn a nhỏ nhất suy ra chỉ tồn tại cung $(v_{a-1}, v_{a+1}) \in E$. Xét các đỉnh v_a và v_{b+1}. Theo cách chọn b lớn nhất suy ra chỉ tồn tại cung $(v_a, v_{b+1}) \in E$.

Khi đó ta có thể xây dựng đường đi có hướng Hamilton khác P sau

$$v_1 \to v_2 \to \dots \to v_{a-1} \to v_{a+1} \to \dots \to v_b \to v_a \to v_{b+1} \to \dots \to v_{n-1} \to v_n$$

- Trường hợp $1 = a < b < n$: Xét các đỉnh v_1 và v_{b+1}. Theo cách chọn b lớn nhất suy ra chỉ tồn tại cung $(v_1, v_{b+1}) \in E$.

Khi đó ta có thể xây dựng đường đi có hướng Hamilton khác P sau

$$v_2 \to \dots \to v_b \to v_1 \to v_{b+1} \to \dots \to v_{n-1} \to v_n$$

- Trường hợp 1 = a < b = n: Ta có chu trình có hướng Hamilton trong G

$$v_1 \to v_2 \to v_3 \to \dots v_{n-1} \to v_n \to v_1$$

và hiển nhiên đồ thị có nhiều đường đi có hướng Hamilton.

□

• ***Định lý 2.4.18 (Moon-Moser).*** Cho G=(V,E) là đồ thị có hướng đủ liên thông mạnh bậc n ($n \geq 3$). Khi đó với mọi đỉnh v và số nguyên p ($3 \leq p \leq n$) luôn tồn tại chu trình có hướng sơ cấp độ dài p qua đỉnh v.

Chứng minh

Cho $v \in V$, ta chứng minh quy nạp theo p. Vì đồ thị liên thông mạnh nên các đỉnh đều có nửa bậc vào và nửa bậc ra dương. Đặt

$$X = \{u \in V \mid \exists (v,u) \in E\}$$

và

$$Y = \{u \in V \mid \exists (u,v) \in E\}$$

Vì G liên thông mạnh nên tồn tại cung từ đỉnh của X đến đỉnh của Y. Suy ra đỉnh v thuộc chu trình độ dài 3.

Giả sử v thuộc chu trình độ dài k, $3 \leq k < n$,

$$C = (v = v_1, v_2, \dots, v_k, v_1)$$

Ta sẽ xây dựng chu trình độ dài k+1 đi qua v. Sẽ xảy ra một trong hai trường hợp sau.

(i) Tồn tại đỉnh $u \notin C$ sao cho có cung đi từ đỉnh nào đó trên C đến u và có cung đi từ u đến đỉnh nào đó trên C. Khi đó sẽ tồn tại hai đỉnh a, $b \in C$, $(a,b) \in E$, sao cho có cung đi từ a đến u và cung đi từ u đến b. Cuối cùng ta có chu trình

$$C' = (a,u,b) \cup C(b,a)$$

qua v và có độ dài k+1.

(ii) Không tồn tại đỉnh u như trường hợp (i). Đặt

$$X = \{x \in V - C \mid \forall z \in C \ \exists \text{ cung } (z, x) \in E\}$$

và

$$Y = \{y \in V - C \mid \forall z \in C \ \exists \text{ cung } (y, z) \in E\}$$

Do (i) nên $X \cup Y = V - C$. Nếu $X = \varnothing$ hay $Y = \varnothing$, thì vi phạm tính liên thông mạnh, vì vậy $X \neq \varnothing$ và $Y \neq \varnothing$.

Cũng do tính liên thông mạnh nên tồn tại $x \in X$ và $y \in Y$ sao cho có cung từ x đến y. Khi đó chu trình

$$C' = (v_1, \dots, v_{k-1}, x, y, v_1)$$

qua đỉnh v và có độ dài k+1.

□

Từ định lý trên suy ra hệ quả sau

• ***Định lý 2.4.19 (Camion).*** Đồ thị có hướng đủ có chu trình có hướng Hamilton khi và chỉ khi nó liên thông mạnh.

5. Mã Gray

• *Định nghĩa 2.4.3*

Dãy 2^n chuỗi *n* bit khác nhau

$$s_1, s_2, \ldots, s_{2^n}$$

gọi là *mã Gray*, nếu thỏa mãn

(i) s_i và s_{i+1} khác nhau đúng 1 bit, i=1, 2, ... , 2^n–1,

(ii) s_1 và s_{2^n} khác nhau đúng 1 bit

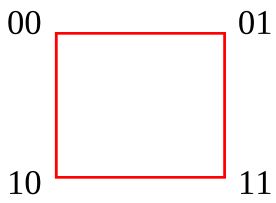
Mã Gray liên quan mật thiết với một loại đồ thị gọi là *siêu khối*. *Siêu khối cấp n* là đồ thị có 2^n đỉnh được gán nhãn 0, 1, 2, ... , $2^n - 1$, trong đó hai đỉnh kề nhau nếu biểu diễn nhị phân *n* bit của chúng chỉ khác nhau đúng 1 bit.

◊ *Ví dụ 2.4.6*

Siêu khối cấp 1

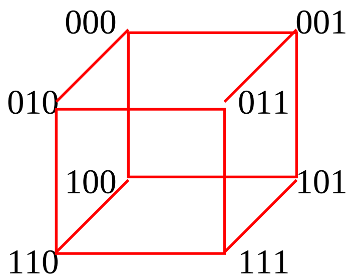
Hình 2.4.14

Siêu khối cấp 2:



Hình 2.4.15

Siêu khối cấp 3:



Hình 2.4.16

Siêu khối được dùng để mô hình hóa máy tính song song và thuật toán song song.

Ta dễ dàng thấy rằng mỗi mã Gray $s_1, s_2, \ldots, s_{2^n}$ tương ứng với một chu trình Hamilton trong siêu khối cấp *n* và ngược lại.

Ngoài ra, mã Gray còn được nghiên cứu ứng dụng trong việc chuyển tính hiệu analog sang tín hiệu số.

• *Định lý 2.4.20.* Ký hiệu G_1 là dãy 0,1. Ta xây dựng truy hồi dãy G_n theo G_{n-1} (n=2,3,...) như sau:

(i) Ký hiệu H_{n-1} là dãy G_{n-1} được viết theo thứ tự đảo ngược.

(ii) Ký hiệu G'_n là dãy các chuỗi *n* bit có được bằng cách thêm bit 0 vào trước các chuỗi của G_{n-1}.

(iii) Ký hiệu G''_n là dãy các chuỗi *n* bit có được bằng cách thêm bit 1 vào trước các chuỗi của H_{n-1}.

(iv) Ký hiệu G_n là dãy các phần tử của G'_n và G''_n nối tiếp nhau.

Khi đó G_n chính là mã Gray với mọi *n*.

Chứng minh. Qui nạp theo n.

• ***Hệ quả 2.4.21.*** Mọi siêu khối cấp $n \geq 2$ có chu trình Hamilton.

V. THUẬT TOÁN NHÁNH CẬN GIẢI BÀI TOÁN NGƯỜI DU LịCH

Thuật toán nhánh cận là một trong các phương pháp chủ yếu giải bài toán tối ưu tổ hợp. Tư tưởng cơ bản của nó là trong quá trình tìm kiếm ta phân hoạch các phương án của bài toán ra thành hai hay nhiều tập con như là các nút của cây tìm kiếm và cố gắng đánh giá cận cho các nút, loại bỏ những nhánh mà ta biết chắc chắn là không chứa phương án tối ưu.

Xét *bài toán người du lịch*. Gọi

$$C = \{ c_{ij} \ : i, j = 1,2,...,n\}$$

là *ma trận chi phí*. Mỗi hành trình

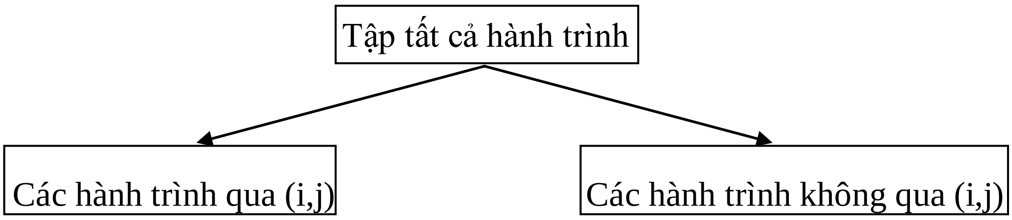
$$v = v(1)\rightarrow v(2)\rightarrow \ldots \rightarrow v(n-1)\rightarrow v(n)\rightarrow v(1)$$

có thể viết dưới dạng

$$v = (v(1),v(2)), (v(2),v(3)),..., (v(n-1),v(n)), (v(n),v(1))$$

trong đó mỗi thành phần $(v(i-1),v(i))$ gọi là một cạnh của hành trình.

Trong bài toán người du lịch khi tiến hành tìm kiếm lời giải ta sẽ phân tập hành trình thành hai tập con: Một tập chứa cạnh (i,j) và tập không chứa cạnh này. Ta gọi việc đó là phân nhánh, mỗi tập con nói trên gọi là nhánh. Việc phân nhánh được minh hoạ bởi cây tìm kiếm:



Việc phân nhánh sẽ được dựa trên qui tắc hợp lý nào đó cho phép ta rút ngắn quá trình tìm kiếm phương án tối ưu. Sau khi phân nhánh ta sẽ tính cận dưới của hàm mục tiêu trong mỗi tập con nói trên. Việc tìm kiếm sẽ tìm trên tập con có cận dưới nhỏ hơn. Thủ tục sẽ tiếp tục cho đến khi thu được hành trình đầy đủ, tức là phương án của bài toán người du lịch. Sau đó ta chỉ cần xét những tập con có cận dưới nhỏ hơn giá trị hàm mục tiêu tìm được.

Kỹ thuật tính *cận dưới* dựa trên thủ tục rút gọn sau.

a. Thủ tục rút gọn

Rõ ràng tổng chi phí của một hành trình sẽ chứa đúng một phần tử trên mỗi dòng và mỗi cột của ma trận chi phí $C = (c_{ij})$. Do đó nếu ta trừ bớt mỗi phần tử của một dòng (hay một cột) đi cùng một giá trị thì chi phí của tất cả hành trình sẽ giảm đi một lượng, vì thế hành trình tối ưu sẽ không thay đổi. Vì vậy nếu tiến hành trừ bớt các phần tử của mỗi dòng và mỗi cột đi một hằng số sao cho thu được ma trận không âm và mỗi cột cũng như mỗi dòng chứa ít nhất một số 0 , thì tổng các hằng số trừ đi đó sẽ cho ta cận dưới của mọi hành trình. Thủ tục trừ bớt này gọi là thủ tục rút gọn, các hằng số trừ ở mỗi dòng (cột) gọi là hằng số rút gọn dòng (cột), ma trận thu được gọi là ma trận rút gọn.

Thủ tục rút gọn:

◊ *Đầu vào* : Ma trận chi phí C = (c_{ij})

◊ *Đầu ra* : Ma trận rút gọn và tổng hằng số rút gọn *Sum*

◊ *Thuật toán*:

(i) Khởi tạo :

Sum := 0 ;

(ii) Rút gọn dòng :

Với mỗi dòng r từ 1 đến n của ma trận C thực hiện :

- Tìm phần tử $c_{rj} = \alpha$ nhỏ nhất trên dòng.
- Trừ tất cả các phần tử trên dòng đi một lượng α.
- Cộng dồn : Sum := Sum + α

(iii) Rút gọn cột :

Với mỗi cột c từ 1 đến n của ma trận C thực hiện :

- Tìm phần tử $c_{ic} = \alpha$ nhỏ nhất trên cột.
- Trừ tất cả các phần tử trên cột đi một lượng α.
- Cộng dồn : Sum := Sum + α

Ví dụ 2.5.1. Cho ma trận chi phí của bài toán người du lịch với *n* = 6 như sau

	1	2	3	4	5	6	α
1	∞	3	93	13	33	9	3
2	4	∞	77	42	21	16	4
3	45	17	∞	36	16	28	16
4	39	90	80	∞	56	7	7
5	28	46	88	33	∞	25	25
6	3	88	18	46	92	∞	3

Ta tiến hành rút gọn ma trận này để tìm cận dưới cho tất cả hành trình :

- *Khởi tạo*:

Đặt Sum := 0;

- *Rút gọn các dòng*:

Các phần tử nhỏ nhất trên các dòng theo thứ tự 1, 2, 3, 4, 5, 6 là 3, 4, 16, 7, 25, 3.

Trừ bớt các phần tử dòng 1, 2, 3, 4, 5, 6 đi các hằng số rút gọn tương ứng trên cột ta được ma trận

	1	2	3	4	5	6
1	∞	0	90	10	30	6
2	0	∞	73	38	17	12
3	29	1	∞	20	0	12
4	32	83	72	∞	49	0
5	3	21	63	8	∞	0
6	0	85	15	43	89	∞
α	0	0	15	8	0	0

Cộng dồn các hằng số rút gọn ta được

$$\text{Sum} := \text{Sum} + 3 + 4 + 16 + 7 + 25 + 3 = 58$$

- *Rút gọn các cột* :

Các phần tử nhỏ nhất trên các cột theo thứ tự 1, 2, 3, 4, 5, 6 là 0, 0, 15, 8, 0, 0.

Trừ bớt các phần tử cột 1, 2, 3, 4, 5, 6 đi các hằng số rút gọn tương ứng trên hàng ta được ma trận rút gọn :

	1	2	3	4	5	6
1	∞	0	75	2	30	6
2	0	∞	58	30	17	12
3	29	1	∞	12	0	12
4	32	83	58	∞	49	0
5	3	21	48	0	∞	0
6	0	85	0	35	89	∞

Cộng dồn các hằng số rút gọn ta được

$$\text{Sum} := \text{Sum} + 0 + 0 + 15 + 8 + 0 + 0 = 58 + 15 + 8 = 81$$

Vậy cận dưới cho tất cả hành trình là

$$\beta = 0 + \text{sum} = 81$$

(nghĩa là không có hành trình có tổng chi phí nhỏ hơn 81).

b. Thủ tục phân nhánh

Giả sử ta chọn cạnh phân nhánh (r, s). Như vậy các hành trình sẽ được chia làm hai tập: P_1 chứa các hành trình qua (r, s) và P_2 chứa các hành trình không qua (r, s).

+ Nhánh tập P_1 : Cận dưới β với giá trị xuất phát có từ thủ tục rút gọn.

- Giảm cấp ma trận chi phí C bằng cách loại dòng r và cột s.
- Ngăn cấm tạo hành trình con :

Cấm cạnh (s,r) bằng cách đặt $c_{sr} = \infty$.

Nếu (r,s) là cạnh phân nhánh thứ hai trở đi thì phải xét các cạnh đã chọn nối trước và sau cạnh (r,s) thành dãy nối tiếp các cạnh như hình sau

$$\underset{(i,j)}{\rightarrow} \quad \dots \quad \underset{(r,s)}{\rightarrow} \quad \dots \quad \underset{(k,h)}{\rightarrow}$$

và cấm tất cả các cạnh dạng (h,i) bằng cách đặt $c_{hi} = \infty$.

Rút gọn ma trận chi phí ta có cận dưới $\beta := \beta +$ (tổng hằng số rút gọn).

Ta có thể tiếp tục thủ tục phân nhánh theo nhánh này với ma trận chi phí đã được hiệu chỉnh và giảm 1 bậc. Việc chọn cạnh nào để phân nhánh ta sẽ bàn ở mục tiếp theo.

+ Nhánh tập P_2 :

- Cấm cạnh (r,s) bằng cách đặt $c_{rs} = \infty$.
- Thực hiện thủ tục rút gọn với ma trận chi phí tương ứng và tính cận dưới $\beta := \beta +$ (tổng hằng số rút gọn) cho nhánh.

Ta có thể tiếp tục thủ tục phân nhánh theo nhánh này với ma trận chi phí đã được hiệu chỉnh cùng cận dưới tương ứng. Việc chọn cạnh nào để phân nhánh ta sẽ bàn ở mục tiếp theo.

Ví dụ 2.5.2. Xét tiếp ví dụ trên. Giả sử ta chọn cạnh phân nhánh là $(r,s) = (6,3)$. Các hành trình sẽ được phân thành hai nhánh:

P_1 chứa các hành trình qua cạnh (6,3) và

P_2 chứa các hành trình không qua cạnh (6,3).

• Xét nhánh tập P_1 :

- Giảm cấp ma trận chi phí C bằng cách loại dòng 6 và cột 3.
- Ngăn cấm tạo hành trình con :

Cấm cạnh (3,6) bằng cách đặt $c_{36} = \infty$.

Ta nhận được ma trận chi phí tương ứng cùng cận dưới xuất phát β = 81.

	1	2	4	5	6
1	∞	0	2	30	6
2	0	∞	30	17	12
3	29	1	12	0	$\underline{\infty}$
4	32	83	∞	49	0
5	3	21	0	∞	0

Vì ma trận đã ở dạng rút gọn nên cận dưới vẫn giữ nguyên $\beta = 81$.

Ta có thể tiếp tục thủ tục phân nhánh theo nhánh này với ma trận chi phí đã được hiệu chỉnh. Việc chọn cạnh nào để phân nhánh ta sẽ bàn ở mục tiếp theo.

• Nhánh tập P_2 :

- Cấm cạnh (6,3) bằng cách đặt $c_{63} = \infty$.
- Thực hiện thủ tục rút gọn với ma trận chi phí tương ứng và tính cận dưới $\beta:=\beta+$(tổng hằng số rút gọn) cho nhánh:

	1	2	3	4	5	6
1	∞	0	75	2	30	6
2	0	∞	58	30	17	12
3	29	1	∞	12	0	12
4	32	83	58	∞	49	0
5	3	21	48	0	∞	0
6	0	85	∞	35	89	∞
α =			48			

Cột thứ 3 có giảm tất cả phần tử đi phần tử nhỏ nhất $c_{53} = 48$ và ta nhận được ma trận rút gọn sau:

	1	2	3	4	5	6
1	∞	0	27	2	30	6
2	0	∞	10	30	17	12
3	29	1	∞	12	0	12
4	32	83	10	∞	49	0
5	3	21	0	0	∞	0
6	0	85	∞	35	89	∞

và cận dưới

$$\beta = \beta + \alpha = 81 + 48 = 129$$

Ta có thể tiếp tục thủ tục phân nhánh theo nhánh này với ma trận chi phí đã được hiệu chỉnh cùng cận dưới tương ứng (129). Việc chọn cạnh nào để phân nhánh ta sẽ bàn ở mục tiếp theo.

c. Thủ tục chọn cạnh phân nhánh

Một cách lôgic là ta chọn cạnh phân nhánh (r,s) sao cho cận dưới của nhánh không chứa (r,s) sẽ tăng nhiều nhất.

Thủ tục chọn cạnh phân nhánh (r,s)

◊ *Đầu vào* : Ma trận rút gọn bậc k

◊ *Đầu ra* : Cạnh phân nhánh (r,s).

◊ *Thuật toán* :

(i) Khởi tạo : $\alpha := -\infty$;

(ii) Với mỗi cặp (i,j) thoả $c_{ij} = 0$ (i=1,...,k; j=1,...,k) thực hiện

- Xác định

$$\text{minr} = \min\{c_{ih} : h \neq j\}$$
$$\text{mins} = \min\{c_{hj} : h \neq i\}$$

- Nếu $\alpha <$ minr + mins , đặt

$$\alpha := \text{minr} + \text{mins}; \quad r := i; \quad s := j;$$

Ví dụ 2.5.3. Xét tiếp ví dụ trên. Ta đi theo nhánh P_1, qua cạnh (6,3). Cận dưới $\beta = 81$ và ma trận chi phí tương ứng như sau

	1	2	4	5	6
1	∞	0	2	30	6
2	0	∞	30	17	12
3	29	1	12	0	∞
4	32	83	∞	49	0
5	3	21	0	∞	0

- Đặt $\alpha = -\infty$.
- Xét các cặp (i,j) có $c_{ij} = 0$.

Cặp (1,2) : minr = 2; mins = 1;
$\alpha = -\infty <$ minr + mins = 3 $\Rightarrow$ $\alpha := 3$; r := 1; s := 2;

Cặp (2,1) : minr = 12; mins = 3;
$\alpha = 3 <$ minr + mins = 15 $\Rightarrow$ $\alpha := 15$; r := 2; s := 1;

Cặp (3,5) : minr = 1; mins = 17;
$\alpha = 15 <$ minr + mins = 18 $\Rightarrow$ $\alpha := 18$; r := 3; s := 5;

Cặp (4,6) : minr = 32; mins = 0;
$\alpha = 18 <$ minr + mins = 32 $\Rightarrow$ $\alpha := 32$; r := 4; s := 6;

Các cặp (5,4) và (5,6) có tổng minr + mins < α, cho nên không làm thay đổi α và (r,s).

Kết quả ta chọn được cạnh phân nhánh là (4,6).

Tập hành trình P_1 sẽ được phân thành 2 nhánh:

P_{11} gồm các hành trình qua cạnh (4,6) và

P_{12} gồm các hành trình không qua cạnh (4,6).

- Nhánh P_{12}: có ma trận chi phí tương ứng như sau

	1	2	4	5	6	
1	∞	0	2	30	6	
2	0	∞	30	17	12	
3	29	1	12	0	∞	
4	32	83	∞	49	∞	32
5	3	21	0	∞	0	
						α

Thực hiện thủ tục rút gọn ta được ma trận rút gọn

	1	2	4	5	6
1	∞	0	2	30	6
2	0	∞	30	17	12
3	29	1	12	0	∞
4	0	51	∞	17	∞
5	3	21	0	∞	0

với tổng hằng số rút gọn $\alpha = 32$ và cận dưới $\beta = \beta + \alpha = 81 + 32 = 113$.

• Nhánh P_{11} :
- Loại hàng 4 và cột 6 ta có sẽ có ma trận chi phí cấp 4 tương ứng:

$$\begin{array}{c|cccc|} & 1 & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 2 & 30 \\ 2 & 0 & \infty & 30 & 17 \\ 3 & 29 & 1 & 12 & 0 \\ 5 & 3 & 21 & 0 & \infty \\ \hline \end{array}$$

- Cấm tạo chu trình con:
Cạnh (6,3) chọn trước kế tiếp sau cạnh (4,6)

$\overrightarrow{(4,6)}\ \overrightarrow{(6,3)}$

Trong bước này ta cấm cạnh (3,4), bằng cách đặt $c_{34} = \infty$. Ta nhận được ma trận

$$\begin{array}{c|cccc|} & 1 & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 2 & 30 \\ 2 & 0 & \infty & 30 & 17 \\ 3 & 29 & 1 & \underline{\infty} & 0 \\ 5 & 3 & 21 & 0 & \infty \\ \hline \end{array}$$

Ma trận chi phí đã ở dạng rút gọn nên cận dưới β giữ nguyên $\beta = 81$.

Bây giờ ta tiếp tục thủ tục chọn cạnh phân nhánh, phân nhánh và rút gọn đối với P_{11}.

Thực hiện thủ tục chọn cạnh phân nhánh, ta chọn được cạnh phân nhánh (2,1) và tổng hằng số rút gọn tương ứng là 17 + 3 = 20.

Tập P_{112} gồm các hành trình trong P_{11} không qua cạnh (2,1) sẽ có cận dưới là $\beta = \beta + 20 = 101$.

Tập P_{111} gồm các hành trình trong P_{11} qua cạnh (2,1) sẽ có ma trận chi phí cấp 3 tương ứng, sau khi đã loại hàng 2 và cột 1:

$$\begin{array}{c|ccc|} & 2 & 4 & 5 \\ \hline 1 & 0 & 2 & 30 \\ 3 & 1 & \infty & 0 \\ 5 & 21 & 0 & \infty \\ \hline \end{array}$$

- Ngăn cấm chu trình con:

Đặt $c_{12} = \infty$ để cấm cạnh (1,2).

Cạnh (2,1) không nối tiếp cùng các cạnh chọn trước là (6,3) và (4,6), nên ta không phải cấm các cạnh khác. Ta có ma trận chi phí

$$
\begin{array}{c|ccc|c}
 & 2 & 4 & 5 & \\
\hline
1 & \underline{\infty} & 2 & 30 & 2 \\
3 & 1 & \infty & 0 & \\
5 & 21 & 0 & \infty & \\
\hline
 & 1 & & & \alpha
\end{array}
$$

Ma trận chi phí được rút gọn như sau

$$\begin{array}{c|ccc|} & 2 & 4 & 5 \\ \hline 1 & \infty & 0 & 28 \\ 3 & 0 & \infty & 0 \\ 5 & 20 & 0 & \infty \\ \hline \end{array}$$

Cận dưới $\beta = \beta + \alpha = 81 + 3 = 84$.

Tương tự ta tiếp tục thủ tục chọn cạnh phân nhánh, phân nhánh và rút gọn đối với P_{111}.

Chọn cạnh phân nhánh (1,4), tổng hằng số rút gọn tương ứng là 28 + 0 = 28.

Tập P_{1112} gồm các hành trình trong P_{111} không qua (1,4) sẽ có cận dưới là $\beta = \beta + 28 = 84 + 28 = 112$.

Tập P_{1111} gồm các hành trình trong P_{111} qua (1,4) sẽ có ma trận chi phí cấp 2 tương ứng:

$$\begin{array}{c|cc|c} & 2 & 5 & \\ \hline 3 & \underline{\infty} & 0 & \\ 5 & 20 & \infty & \\ \hline & 20 & & \alpha \end{array}$$

Cạnh (1,4) nối tiếp cùng các cạnh chọn trước là (6,3) , (4,6) và (2,1) thành dãy

(2,1),(1,4),(4,6),(6,3) ,

nên ta phải cấm cạnh (3,2), bằng cách đặt $c_{32} = \infty$.

Rút gọn ma trận này ta được ma trận rút gọn

$$\begin{array}{c|cc|} & 2 & 5 \\ \hline 3 & \infty & 0 \\ 5 & 0 & \infty \\ \hline \end{array}$$

và cận dưới $\beta = \beta + \alpha = 84 + 20 = 104$.

d. Chọn 2 cạnh cuối cùng

Mỗi hành trình có n cạnh. Sau khi đã chọn n–2 cạnh, ta phải chọn nốt 2 cạnh còn lại. Lúc này ma trận rút gọn có bậc 2 và là một trong hai dạng sau

$$\begin{array}{c|cc|} & u & v \\ \hline p & 0 & \infty \\ q & \infty & 0 \\ \hline \end{array} \qquad\qquad \begin{array}{c|cc|} & u & v \\ \hline p & \infty & 0 \\ q & 0 & \infty \\ \hline \end{array}$$

(i) (ii)

cùng cận dưới β.

Trong trường hợp (i) ta chọn hai cạnh (p,u) và (q,v), còn trong trường hợp (ii) ta chọn hai cạnh (p,v) và (q,u). Tổng chi phí là β.

Ví dụ 2.5.4. Xét tiếp ví dụ trên. Sau khi chọn các cạnh (6,3),(4,6),(2,1),(1,4) ta có ma trận rút gọn

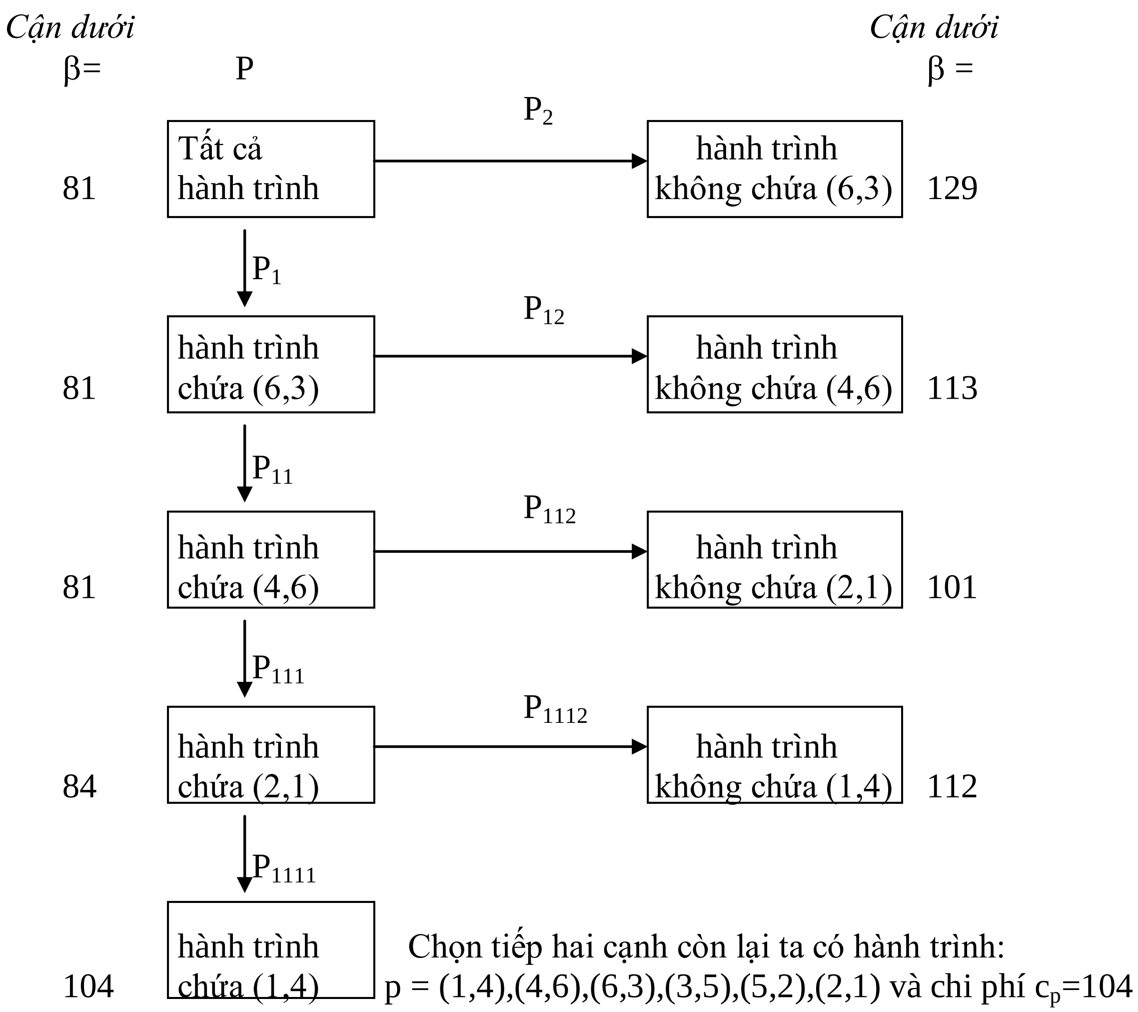
	2	5
3	∞	0
5	0	∞

và cận dưới β = 104. Ta chọn nốt hai cạnh còn lại là (3,5) và (5,2) và được hành trình

(1,4),(4,6),(6,3),(3,5),(5,2),(2,1)

với tổng chi phí bằng β = 104.

Quá trình trên có thể biểu diễn bằng sơ đồ sau:



Qua sơ đồ trên ta thấy các nhánh P_2 , P_{12} , P_{1112} có cận dưới lớn hơn c_p = 104, vì thế các hành trình của các nhánh đó có tổng chi phí lớn hơn c_p. Chỉ có nhánh P_{112} có cận dưới là 101 nhỏ hơn c_p. Tiếp theo ta tìm hành trình mới theo nhánh này.

- Nhánh P_{112} sẽ có ma trận chi phí cấp 4 tương ứng:

$$\begin{array}{c|cccc|c}
 & 1 & 2 & 4 & 5 & \\ \hline
1 & \infty & 0 & 2 & 30 & \\
2 & \infty & \infty & 30 & 17 & 17 \\
3 & 29 & 1 & \infty & 0 & \\
5 & 3 & 21 & 0 & \infty & \\ \hline
 & 3 & & & & \alpha
\end{array}$$

ở đây $c_{21} = \infty$ để cấm cạnh (2,1). Cận dưới xuất phát của P_{11} là $\beta = 81$.

Rút gọn ma trận chi phí ta được ma trận

$$\begin{array}{c|cccc|}
 & 1 & 2 & 4 & 5 \\ \hline
1 & \infty & 0 & 2 & 30 \\
2 & \infty & \infty & 13 & 0 \\
3 & 26 & 1 & \infty & 0 \\
5 & 0 & 21 & 0 & \infty \\ \hline
\end{array}$$

với cận dưới $\beta = \beta + \alpha = 81 + 20 = 101$.

Chọn (5,1) làm cạnh phân nhánh. Nhánh P_{1122} gồm các hành trình không qua cạnh (5,1) có cận dưới bằng $101 + 26 = 127 > 104 = c_p$. Ta loại không xét nhánh này nữa.

- Nhánh P_{1121} gồm các hành trình qua cạnh (5,1) có ma trận chi phí

$$\begin{array}{c|ccc|c}
 & 2 & 4 & 5 & \\ \hline
1 & 0 & 2 & \infty & \\
2 & \infty & 13 & 0 & \\
3 & 1 & \infty & 0 & \\ \hline
 & & 2 & & \alpha
\end{array}$$

($c_{15} = \infty$ để cấm cạnh (1,5)).

Rút gọn ma trận chi phí ta có ma trận rút gọn

$$\begin{array}{c|ccc|}
 & 2 & 4 & 5 \\ \hline
1 & 0 & 0 & \infty \\
2 & \infty & 11 & 0 \\
3 & 1 & \infty & 0 \\ \hline
\end{array}$$

với cận dưới $\beta = \beta + ỏ = 101 + 2 = 103$.

Chọn (1,4) làm cạnh phân nhánh. Nhánh P_{11212} gồm các hành trình không qua cạnh (1,4) có cận dưới bằng $103 + 11 = 114 > 104 = c_p$. Ta loại không xét nhánh này nữa.

- Nhánh P_{11211} gồm các hành trình qua cạnh (1,4) có ma trận chi phí

$$\begin{array}{c|cc|c} & 2 & 5 & \\ \hline 2 & \infty & 0 & \\ 3 & 1 & \underline{\infty} & 1 \\ \hline & & & \alpha \end{array}$$

Cạnh (1,4) cùng với các cạnh đã chọn tạo thành dây

(5,1), (1,4), (4,6), (6,3)

vì thế ta cấm cạnh (3,5), bằng cách đặt $c_{35} = \infty$. Rút gọn ta được ma trận

$$\begin{array}{c|cc|} & 2 & 5 \\ \hline 2 & \infty & 0 \\ 3 & 0 & \infty \\ \hline \end{array}$$

và cận dưới $\beta = 103 + 1 = 104$.

Chọn tiếp 2 cạnh còn lại (3,2) và (2,5) ta được hành trình

(1,4),(4,6),(6,3),(3,2),(2,5),(5,1)

với tổng chi phí là 104.

- *Kết luận*: Hai hành trình tìm được là tối ưu với chi phí thấp nhất là 104:

(1,4),(4,6),(6,3),(3,5),(5,2),(2,1)

và

(1,4),(4,6),(6,3),(3,2),(2,5),(5,1)

BÀI TẬP

I. ĐƯỜNG ĐI EULER

2.1.1. Xác định điều kiện của tham số *n* để đồ thị K_n có chu trình Euler, đường đi Euler.

2.1.2. Xác định điều kiện của các tham số *m*, *n* để đồ thị $K_{m,n}$ có chu trình Euler, đường đi Euler.

2.1.3. Các đồ thị vô hướng cho bởi ma trận kề sau có chu trình hay đường đi Euler hay không. Nếu có hãy chỉ ra các chu trình hay đường đi Euler.

(**a**)

$$\begin{pmatrix}
 & 1 & 1 & 1 & 1 & & \\
1 & & & 1 & & & \\
1 & & & 1 & & 1 & 1 \\
1 & 1 & 1 & & 1 & 1 & 1 \\
1 & & & 1 & & 1 & 1 \\
 & & 1 & 1 & 1 & & 1 \\
 & & 1 & 1 & 1 & 1 &
\end{pmatrix}$$

(**b**)

$$\begin{pmatrix}
 & 1 & 1 & & 1 & 1 & \\
1 & & 1 & & & 1 & \\
1 & 1 & & 1 & & 1 & \\
 & & 1 & & 2 & & 1 \\
1 & & & 2 & & 1 & \\
1 & 1 & 1 & & 1 & & \\
 & & & 1 & & &
\end{pmatrix}$$

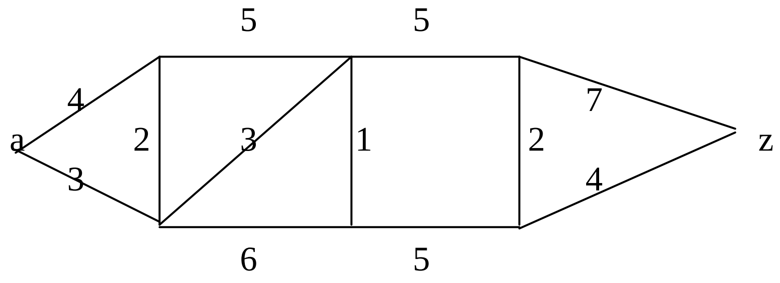
II. TÌM ĐƯỜNG ĐI TRONG MÊ CUNG

2.2.1. Tìm tất cả lời giải bài toán sói dê cải.
2.2.2. Tìm tất cả lời giải bài toán ba ông chồng ghen.
2.2.3. Tìm tất cả lời giải bài toán ba thầy tu và ba con quỷ..

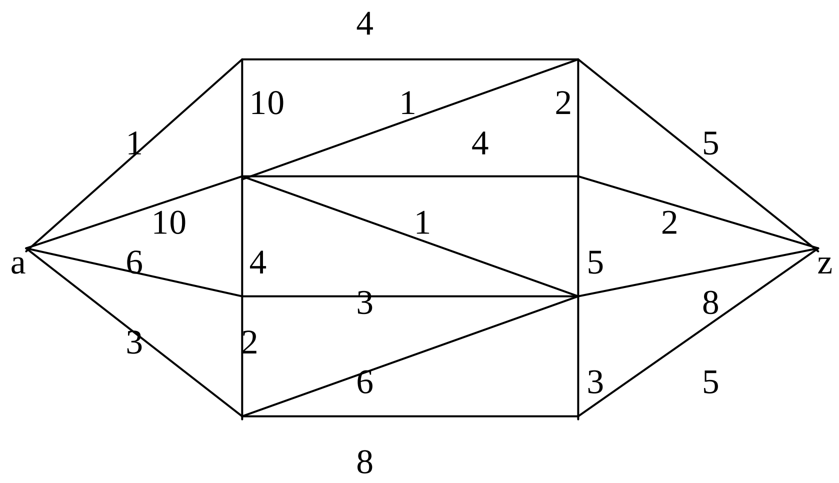
III. TÌM ĐƯỜNG ĐI NGẮN NHẤT

2.3.1. Dùng giải thuật Dijkstra tìm đường đi ngắn nhất từ đỉnh a đến đỉnh z trong các đồ thị có trọng số sau

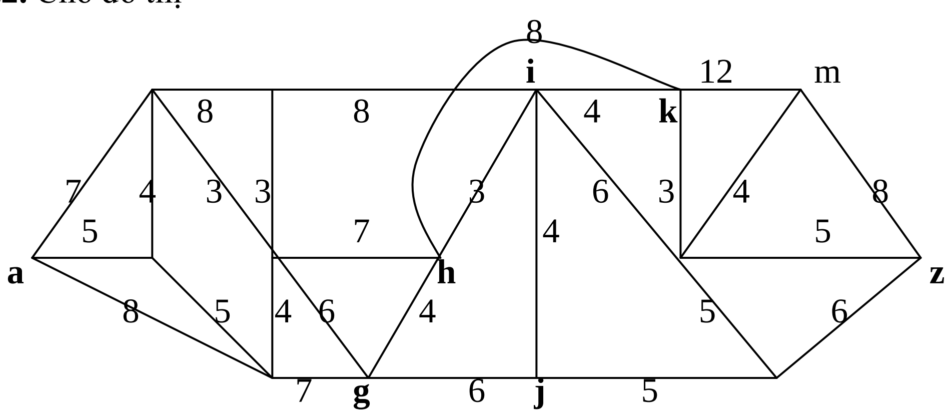
(**a**)



(**b**)



2.3.2. Cho đồ thị



Dùng giải thuật Dijkstra tìm đường đi ngắn nhất

(**a**) từ đỉnh **a** đến đỉnh **z**

(**b**) từ đỉnh **a** đến đỉnh **z** và qua cạnh (**h,g**)

(**c**) từ đỉnh **a** đến đỉnh **z** và qua đỉnh **k**

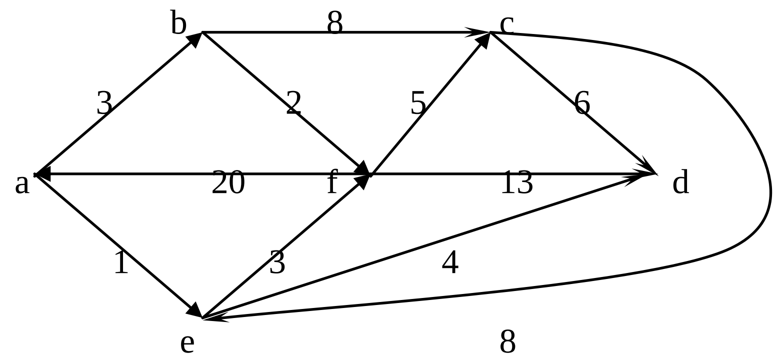
(**d**) từ đỉnh **a** đến đỉnh **z** và qua cạnh (**i,j**)

2.3.3. Dùng giải thuật Bellman-Ford tìm đường đi ngắn nhất giữa các đỉnh trong các đồ thị có trọng số ở bài **2.3.1, 2.3.2.**

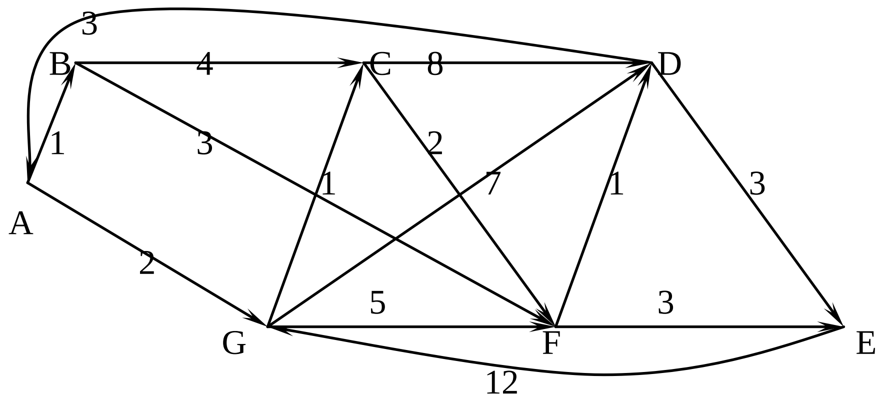
2.3.4. Dùng giải thuật Floyd-Warshall tìm đường đi ngắn nhất giữa các đỉnh trong các đồ thị có trọng số ở bài **2.3.1.**

2.3.5. Dùng giải thuật Floyd-Warshall tìm đường đi ngắn nhất giữa các đỉnh trong các đồ thị có hướng có trọng số sau

(**a**)



(**b**)



IV. ĐƯỜNG ĐI HAMILTON

2.4.1. Tìm đồ thị G thỏa mãn

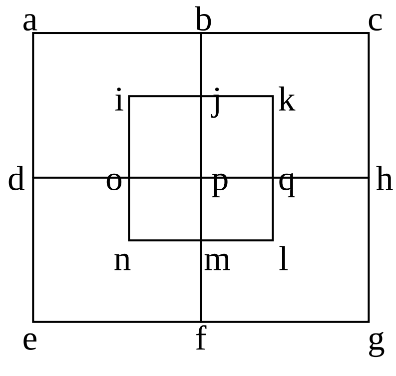
(**a**) có chu trình Hamilton, nhưng không có đường đi Euler.

(**b**) có chu trình Euler, nhưng không có đường đi Hamilton.

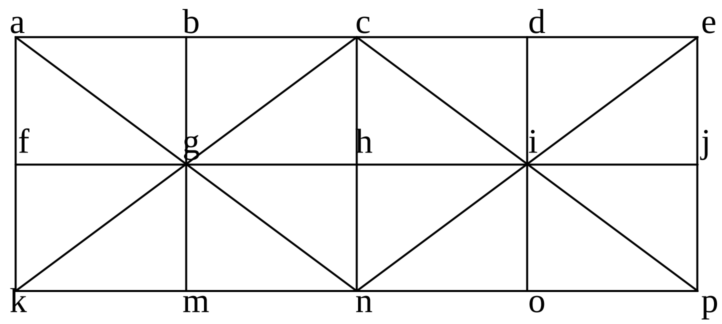
2.4.2. Xác định điều kiện của các tham số *m, n* để đồ thị $K_{m,n}$ có chu trình Hamilton, đường đi Hamilton.

2.4.3. Các đồ thị sau có chu trình hay đường đi Hamilton không. Nếu có hãy chỉ ra chu trình hay đường đi Hamilton.

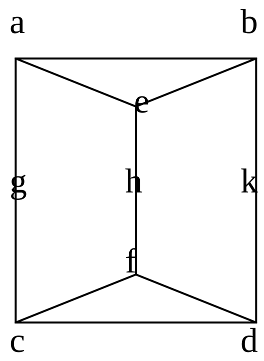
(**a**)



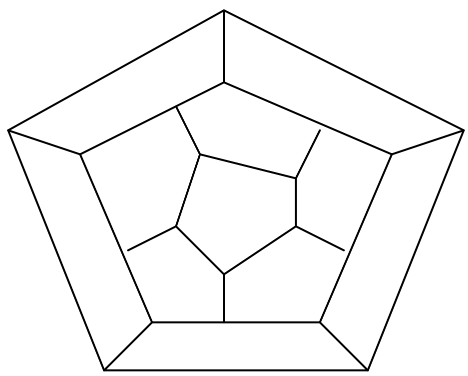
(**b**)



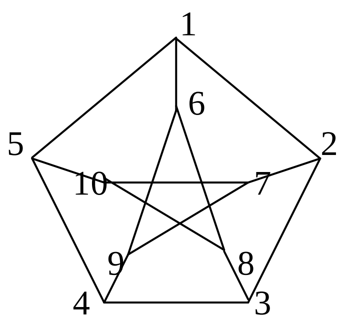
(**c**)



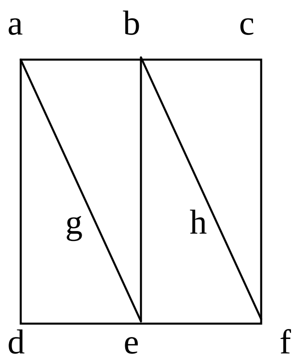
(**d**)



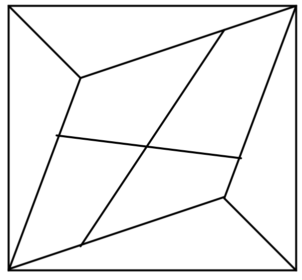
(**e**) Đồ thị Peterson



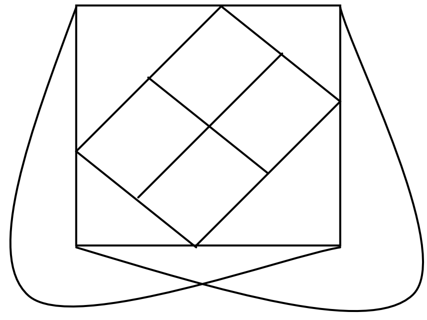
(**f**)



(**g**)



(**h**)



2.4.4. Có n người, trong đó mỗi người quen ít nhất $\lceil n/2 \rceil$ ($\lceil x \rceil$ là số nguyên nhỏ nhất $\geq x$) người. Hỏi có thể xếp n người này vào bàn tròn sao cho mỗi người ngồi cạnh hai người quen ?

2.4.5. Có n nam và n nữ, trong đó mỗi người quen ít nhất $\lceil n/2 + 1 \rceil$ người khác giới. Hỏi có thể xếp xen kẽ nam nữ $2n$ người này vào bàn tròn sao cho mỗi người ngồi cạnh hai người quen ?

BÀI TẬP THỰC HÀNH

B.2.1. Viết chương trình cài đặt thủ tục tìm chu trình Euler.

◊ *Hướng dẫn*

Giả sử G là đồ thị Euler, từ thuật toán 2.1.1 ta xây dựng thủ tục sau để tìm chu trình Euler trong G. Trong thủ tục này *CE* là chu trình Euler (mảng hoặc danh sách liên kết), *STACK* là ngăn xếp, *EList* là danh sách cạnh.

```
Procedure Euler_Cycle;
Begin
    STACK :=∅ ; CE :=∅ ;
    Chọn u là một đỉnh nào đó của đồ thị;
    u ⇒ STACK; {*đẩy u vào ngăn xếp*}
    While STACK<>∅ do
        Begin
            x:= top(STACK); {* x là phần tử đầu STACK *}
            If ∃(x,y) ∈ Elist then
                begin
                    y ⇒ STACK;
                    {* loại bỏ cạnh (x,y) khỏi đồ thị *}
                    EList :=Elist \ {(x, y)} ;
                end
            Else
                begin
                    STACK ⇒ x; x ⇒ CE;
                end;
        End;
End;
```

B.2.2. Viết chương trình cài đặt thủ tục tìm số đường đi tối thiểu phủ đồ thị của đồ thị G.

◊ *Hướng dẫn*

1) Xác định $2k$ đỉnh bậc lẻ của đồ thị G.

2) Thêm k cạnh ảo nối k cặp đỉnh bậc lẻ của G, ký hiệu đồ thị nhận được là G'.

3) Tìm chu trình Euler C' của G'.

4) Loại các cạnh ảo trên chu trình C', nhận được k đường đi tối thiểu phủ đồ thị G.

B.2.3. Cài đặt thuật toán giải bài toán người, sói, dê và cải.

◊ *Hướng dẫn cài đặt*

Mỗi nút biểu diễn bằng bản ghi có 4 thành phần (n,s,d,c) với quy ước

n = 1 nghĩa là có người, n = 0 nghĩa là không có người

s = 1 nghĩa là có sói, s = 0 nghĩa là không có sói

d = 1 nghĩa là có dê, d = 0 nghĩa là không có dê

c = 1 nghĩa là có cải, c = 0 nghĩa là không có cải

Nút (n,s,d,c) là chấp nhận khi và chỉ khi (n = 1 & s+d>0 & d+c>0) hoặc (n = 0 & s.d = 0 & d.c = 0).

Hai nút chấp nhận (n_1,s_1,d_1,c_1) và (n_2,s_2,d_2,c_2) kề nhau khi và chỉ khi

$$n_1 + n_2 = 1 \ \& \ |\, (s_1+d_1+c_1) - (s_2+d_2+c_2)\, | \le 1$$

B.2.4. Cài đặt thuật toán giải bài toán ba ông chồng ghen.

◊ *Hướng dẫn cài đặt*

Mỗi nút biểu diễn bằng bản ghi có 6 thành phần (A,a,B,b,C,c) với quy ước rằng với mọi $t \in \{A,a,B,b,C,c\}$,

t = 1 (*true*) nghĩa là t có mặt, t = 0 (*false*) nghĩa là t không có mặt

Nút (A’,a’,B’,b’,C’,c’) gọi là *bù* của (A,a,B,b,C,c) nếu

(A = not A’) and (a = not a’) and (B = not B’) and
(b = not b’) and (C = not C’) and (c = not c’)

Nút (A,a,B,b,C,c) là chấp nhận nó và nút bù thỏa điều kiện

(a and (B or C) ⇒ A) and (b and (A or C) ⇒ B) and (c and (B or A) ⇒ C) (*)

Hai nút chấp nhận (A,a,B,b,C,c) và (A’,a’,B’,b’,C’,c’) kề nhau khi và chỉ khi 6 số (A−A’), (a−a’), (B−B’), (b−b’), (C−C’), (C−c’) cùng dấu và

$$1 \le |A-A'|+|a-a'|+|B-B'|+|b-b'|+|C-C'|+|C-c'| \le 2$$

B.2.5. Cài đặt thuật toán giải bài toán ba thầy tu và ba con quỷ.

◊ *Hướng dẫn cài đặt*

Ký hiệu nút trạng thái ở bờ sông là (n,m), trong đó n là số thầy tu, m là số quỷ.

Nút (n’,m’) gọi là bù của nút (n,m) nếu n+n’=3 và m+m’=3.

Nút (n,m) là chấp nhận nếu nó và nút bù (n’,m’) thỏa

$$(0 \le n,\ m \le 3)\ \&\ [(n = 0) \text{ or } (n \ge m)]$$

Hai nút chấp nhận (n, m), (n’,m’) kề nhau nếu (n−n’) và (m−m’) cùng dấu và

$$1 \le |n-n'| + |m-m'| \le 2$$

B.2.6. Viết chương trình cài đặt thuật toán Dijkstra tìm đường đi ngắn nhất giữa hai đỉnh.

Hướng dẫn

```
Procedure Dijkstra;
(*
    Đầu vào:
        Đồ thị có hướng G=(V,E) với n đỉnh,
        s ∈ V là đỉnh xuất phát, t∈V là đỉnh kết thúc, a[u,v], u,v ∈ V, ma trận trọng số;
        Giả thiết: a[u,v] ≥ 0, u,v ∈ V.
    Đầu ra:
        Khoảng cách d[t] từ đỉnh s đến t
        ( d[v] khoảng cách từ s đến v, v ∈ V).
       Truoc[v], v∈V, ghi nhận đỉnh đứng trước v trong đường đi ngắn nhất từ s đến v
*)
Begin
    (* Khởi tạo *)
    for v ∈ V do
```

```
        begin
            d[v]:=a[s,v];
            Truoc[v]:=s;
            Chon[v] := false;
        end;
    d[s]:=0; chon[s] := true; u := s;
    (* Bước lặp *)
    repeat
        For v∈ V thỏa not chon[v] do
            If d[v]>d[u]+a[u,v] then
                Begin
                    d[v]:=d[u]+a[u,v];
                    Truoc[v]:=u;
                End;
        {Tìm đỉnh u ∈ V  thoả mãn d[u]=min{ d[z]:z ∈ V và not chon[z]} ;}
        min := oo;{oo  là biến lưu giá trị lớn nhất của kiểu dữ liệu trọng số}
        for i := 1 to n do
             if not chon[i] and d[i] < min then
                 begin
                    min := d[i];
                    u := i;
                 end;
        if min < oo then
            chon[u]:= true ; (* Cố định nhãn của đỉnh u*)
    until chon[t] or min = oo {Không tồn tại đường đi từ s đến t};
End;
```

B.2.7. Viết chương trình cài đặt thuật toán Bellman-Ford tìm đường đi ngắn nhất giữa hai đỉnh.

B.2.8. Viết chương trình cài đặt thuật toán Floyd-Warshall tìm đường đi ngắn nhất giữa mọi cặp đỉnh.

B.2.9. Viết chương trình cài đặt thuật toán quay lui tìm chu trình Hamilton.

Hướng dẫn

Thuật toán quay lui liệt kê tất cả các chu trình Hamilton của đồ thị

Thủ tục sau đây được xây dựng dựa trên cơ sở thuật toán quay lui cho phép liệt kê tất cả các chu trình Hamilton của đồ thị.

```
Procedure Hamilton(k);
(* liệt kê các chu trình Hamilton  thu được bằng việc phát triển dãy đỉnh (X[0]=v0,. . . ,
X[k–1]) của đồ thị G=(V,E) có n đỉnh cho bởi danh sách kề: Ke(v), v∈V *)
begin
    for y ∈ Ke(X[k–1]) do
        if (k = n) and (y=v0) then Ghinhan(X[0],. . . , X[n–1], v0)
        else
            if (k <n) and Chuaxet[y] then
                begin
                    X[k]:=y;
                    Chuaxet[y]:=false;
                    Hamilton(k+1);
                    Chuaxet[y]:=true;
```

```
                end;
end;
(* Main program*)
begin
    for v ∈ V do Chuaxet[v]:=true;
    X[0]:=v0; (* v0 là một đỉnh nào đó của đồ thị *)
    Chuaxet[v0]:=false;
    Hamilton(1);
end.
```

B.2.10. Viết chương trình cài đặt thuật toán xây dựng mã Gray.

B.2.11. Viết chương trình cài đặt thuật toán nhánh cận giải bài toán người du lịch.