

## **BÀI 14**

### **Chương 8** **Bài toán đường đi ngắn nhất**

Trước mỗi chuyến xuất hành, chúng ta thường phải suy nghĩ và chọn ra cho mình một hành trình “tiết kiệm” nhất theo nghĩa tốn ít thời gian, tốn ít nhiên liệu hoặc tốn ít tiền nhất ... Lý thuyết Đồ thị sẽ giúp chúng ta tìm ra giải pháp đó.

#### 8.1. Bài toán Đường đi ngắn nhất

**Bài toán:** Cho đồ thị  $G = (V, E)$  và hai đỉnh  $a, b$ . Tìm đường đi ngắn nhất (nếu có) đi từ đỉnh  $a$  đến đỉnh  $b$  trong đồ thị  $G$ .

ý nghĩa thực tế: Bài toán này giúp chúng ta chọn các hành trình tiết kiệm nhất (quãng đường, thời gian, chi phí ...) trong giao thông, lập lịch thi công các công trình một cách tối ưu, xử lý trong truyền tin ...

Thuật toán duyệt đồ thị theo chiều rộng đã cho ta lời giải của bài toán này. Song ta có thêm thuật toán sau đây.

#### **Thuật toán 8.1:**

1. Lần lượt gán nhãn cho các đỉnh của đồ thị, mỗi đỉnh không quá một lần, như sau:

- Đỉnh  $a$  được gán nhãn là số 0.
- Những đỉnh kề với đỉnh  $a$  được gán số 1.
- Những đỉnh kề với đỉnh đã được gán nhãn số 1, được gán số 2.

.....  
- Tương tự, những đỉnh kề với đỉnh đã được gán số  $i$  được gán nhãn là số  $i+1$ .

.....  
Thực hiện cho đến khi gán được nhãn cho đỉnh  $b$  hoặc không gán nhãn được nữa.

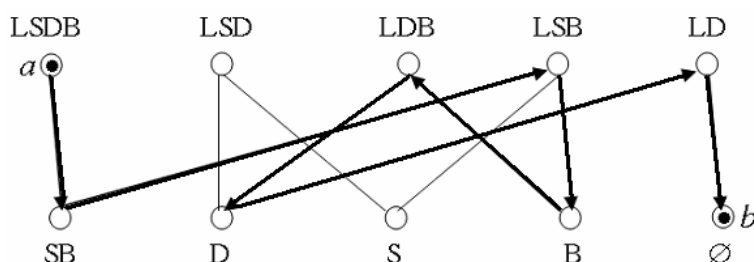
2. Nếu đỉnh  $b$  được gán nhãn nào đó là  $k$  thì kết luận có đường đi ngắn nhất từ đỉnh  $a$  tới đỉnh  $b$  với độ dài  $k$ , ngược lại thì trả lời là không có.

3. *Khôi phục đường đi:* Nếu ở bước 2. chỉ ra  $b$  được gán nhãn  $k$  nào đó thì ta đi ngược lại theo quy tắc sau đây: Nếu đỉnh  $y$  được gán nhãn  $j$  với  $j \geq 1$  thì sẽ có đỉnh  $x$  được gán nhãn  $j-1$  sao cho có cạnh đi từ  $x$  tới  $y$ . Đi ngược lại cho đến khi gặp đỉnh  $a$ , ta nhận được đường đi ngắn nhất cần tìm.

Ví dụ 8.1 (*Bài toán con sói, con dê và cái bắp cải*):

Một con sói, một con dê và một cái bắp cải đang ở bờ sông. Người lái đò phải đưa chúng sang sông. Nhưng thuyền quá bé nên mỗi chuyến chỉ chở được một “hành khách” thôi. Vì những lý do mà ai cũng biết, không thể bỏ mặc sói với dê hoặc dê với bắp cải mà không có người trông. Vậy người lái đò phải xử trí thế nào mà vẫn đưa được sói, dê và bắp cải sang bên kia sông.

Xây dựng đồ thị vô hướng với các đỉnh thể hiện các hành khách còn lại bên phía xuất phát tại mỗi thời điểm khác nhau. Cạnh nối hai đỉnh thể hiện một chuyến đò qua sông.



Hình 8.1. Hành trình qua sông của sói, dê và bắp cải

Bài toán đưa về việc tìm đường đi ngắn nhất từ đỉnh  $a$  đến đỉnh  $b$  trên đồ thị. Đường đi như thế được chỉ ra bởi các mũi tên ở hình trên.

## 8.2. Bài toán Đường đi có trọng số bé nhất

Với bài toán đường đi tổng quát, ta xét các đồ thị có trọng số được định nghĩa như sau.

### **Định nghĩa 8.2:**

Đồ thị  $G$  được gọi là *đồ thị có trọng số* nếu trên mỗi cạnh  $(i, j)$  của đồ thị được gán một số nguyên không âm  $c(i, j)$ .

Nhãn  $c(i, j)$  trên cạnh  $(i, j)$  của đồ thị thường biểu diễn “chi phí” thực tế để đi qua cạnh này.

Ta thường ký hiệu đồ thị có trọng số là  $(G, c)$ .

Độ dài của đường đi trong đồ thị có trọng số bằng tổng các trọng số của các cạnh trên đường đi đó.

**Bài toán:** Cho đồ thị có trọng số  $(G, c)$  và hai đỉnh  $a, b$  thuộc  $G$ . Hãy tìm đường đi có trọng số bé nhất (nếu có) đi từ đỉnh  $a$  đến đỉnh  $b$ .

Độ dài đường đi ngắn nhất từ đỉnh  $a$  đến đỉnh  $b$  còn được gọi là *khoảng cách* từ đỉnh  $a$  đến đỉnh  $b$  trong đồ thị. Nếu không có đường đi từ  $a$  đến  $b$  thì đặt khoảng cách bằng  $\infty$ .

### 8.3. Thuật toán Dijkstra tìm đường đi ngắn nhất

Năm 1959 E. W. Dijkstra đưa ra một thuật toán rất hiệu quả để giải bài toán đường đi ngắn nhất.

Thuật toán thực hiện việc gán và giảm giá trị của nhãn  $l(i)$  tại mỗi đỉnh  $i$  của đồ thị  $G$  như sau:

**Thuật toán 8.2** (Tìm đường đi ngắn nhất):

1. Với đỉnh xuất phát  $a$ , gán nhãn  $l(a) := 0$ .
2. Nếu có cạnh  $(i, j)$  mà đỉnh  $i$  đã được gán nhãn và đỉnh  $j$  chưa được gán nhãn hoặc đỉnh  $j$  đã được gán nhãn nhưng  $l(i) + c(i, j) < l(j)$  thì giảm nhãn  $l(j) := l(i) + c(i, j)$ .
3. Lặp lại bước 2. cho đến khi không gán hoặc giảm nhãn được nữa.

**Định lý 8.3:** Tại mỗi đỉnh  $b$  giá trị nhãn  $l(b)$  cuối cùng (nếu có) chính là độ dài của đường đi ngắn nhất từ đỉnh  $a$  đến đỉnh  $b$ .

Chứng minh:

Sau khi đã thực hiện xong thuật toán trên, nếu giá trị nhãn  $l(b)$  xác định thì ta có đường đi từ đỉnh  $a$  tới đỉnh  $b$ .

Ta khôi phục đường đi từ  $a$  đến  $b$  như sau:

Xuất phát từ đỉnh  $b$ , tìm cạnh có đỉnh cuối là  $b$  và đỉnh đầu là  $i$  sao cho:

$$l(i) + c(i, b) = l(b).$$

Đỉnh  $i$  như thế chắc chắn phải tồn tại vì xảy ra đẳng thức ở lần gán hoặc giảm giá trị nhãn  $l(j)$  cuối cùng. Cứ tiếp tục như thế cho đến khi gặp đỉnh  $a$ .

Giả sử ta nhận được dãy các cạnh:

$$(a, a_1), (a_1, a_2), \dots, (a_{k-1}, b)$$

$$\text{mà trên đó: } l(a) + c(a, a_1) = l(a_1)$$

$$l(a_1) + c(a_1, a_2) = l(a_2)$$

$$\dots \dots \dots$$

$$l(a_{k-1}) + c(a_{k-1}, b) = l(b).$$

Cộng từng vế và khử các giá trị chung ở cả hai vế ta có:

$$c(a, a_1) + c(a_1, a_2) + \dots + c(a_{k-1}, b) = l(b).$$

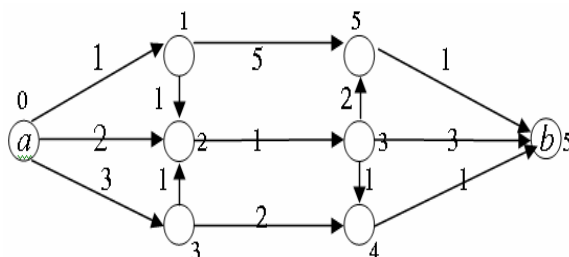
Vậy giá trị nhãn  $l(b)$  chính là độ dài đường đi nói trên.

Bất kỳ đường đi nào khác từ đỉnh  $a$  đến đỉnh  $b$  cũng có các hệ thức tương tự nhưng có dấu  $\geq$ .

Vậy nhãn  $l(b)$  là độ dài của đường đi ngắn nhất.

□

Ví dụ 8.3: Xét đồ thị có trọng số sau đây:



Hình 8.2. Đồ thị có trọng số

Độ dài đường đi ngắn nhất từ đỉnh  $a$  đến đỉnh  $b$  là 5.

Để đơn giản việc tính toán, ta xây dựng ma trận trọng số  $C$ :

$$C[i,j] = \begin{cases} c(i,j) & , \text{ nếu } (i,j) \in E \\ \infty & , \text{ nếu } (i,j) \notin E \\ 0 & , \text{ nếu } i = j. \end{cases}$$

Khi đó, thuật toán Dijkstra được trình bày chi tiết hơn như sau:

```

1 procedure DIJKSTRA( $a$ ) ;
2 begin
3   for  $j \in V$  do
4     begin
5        $L[j] := C[a, j]$  ;  $Truoc[j] := a$ 
6     end ;
7    $T := V \setminus \{a\}$  ;
8   while  $T \neq \emptyset$  do
9     begin
10      chọn đỉnh  $i \in T$  mà  $L[i] = \min \{L[j] \mid j \in T\}$  ;
11       $T := T \setminus \{i\}$  ;
12      for  $j \in T$  do
13        if  $L[j] > L[i] + C[i, j]$  then
14          begin
15             $L[j] := L[i] + C[i, j]$  ;
16             $Truoc[j] := i$  ;

```

```

17           end ;
18     end ;
19 end ;

```

Biến mảng Truoc dùng để khôi phục đường đi.

#### 8.4. Đường đi trên đồ thị phi chu trình

Sử dụng Thuật toán 4.5 (Chương 4) để đánh số các đỉnh trên đồ thị định hướng phi chu trình, ta xây dựng được thuật toán ngắn gọn hơn để tìm khoảng cách từ đỉnh nguồn tới tất cả các đỉnh trong một đồ thị phi chu trình.

##### **Thuật toán 8.4:**

*Dữ liệu:* Biểu diễn mảng DK\_V các danh sách kề của đồ thị định hướng phi chu trình  $G = (V, E)$  với tập đỉnh  $V = \{v_1, v_2, \dots, v_n\}$  đã được đánh số mà danh sách DK\_V[ $v_j$ ] chứa các đỉnh nhận  $v_j$  là đỉnh kề và ma trận trọng số C của đồ thị G.

*Kết quả:* Mảng D các số nguyên với D[ $v_i$ ] chứa khoảng cách  $d(v_1, v_i)$ ,  $i = 2, 3, \dots, n$ .

```

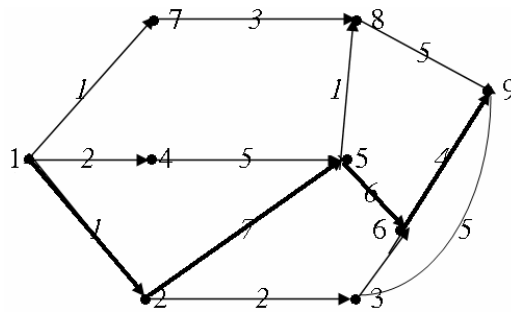
1 Begin
2   D[ $v_1$ ] := 0 ;
3   for  $j := 2$  to  $n$  do D[ $v_j$ ] :=  $\infty$  ;
4   for  $j := 2$  to  $n$  do
5     for  $v_i \in \text{DK\_V}[v_j]$  do D[ $v_j$ ] := min( D[ $v_j$ ] , D[ $v_i$ ] + C[ $v_i, v_j$ ]
6   End.

```

Tính đúng đắn của thuật toán suy từ chi tiết sau đây: tất cả các đỉnh trung gian trên đường đi ngắn nhất từ  $v_1$  tới  $v_j$  có chỉ số nhỏ hơn  $j$ . Mỗi cạnh  $(v_i, v_j)$  được xét trong dòng lệnh 5 đúng một lần, do vậy độ phức tạp của thuật toán là  $O(m)$ .

Ta cũng có thể áp dụng thuật toán trên để tìm đường đi dài nhất từ đỉnh nguồn tới các đỉnh khác của đồ thị hoặc tìm đường đi dài nhất trên đồ thị định hướng phi chu trình có trọng số.

Ví dụ 8.4: Tìm đường đi dài nhất trên đồ thị định hướng phi chu trình có trọng số dưới đây.



Hình 8.3. Đường đi dài nhất trên đồ thị phi chu trình có trọng số

### 8.5. Đường đi ngắn nhất giữa tất cả các cặp đỉnh

**Bài toán:** Cho một đồ thị có trọng số  $(G, c)$ . Hãy tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh.

Bài toán này thường gặp trong việc xây dựng bảng khoảng cách giữa các thành phố, bảng giá cước vận chuyển giữa các nhà ga ...

Bài toán này có thể giải quyết bằng cách sử dụng thuật toán Dijkstra với mỗi đỉnh của đồ thị lần lượt là các đỉnh xuất phát. Tuy nhiên, ta có thể giải quyết trực tiếp bài toán nhờ thuật toán Floyd như sau:

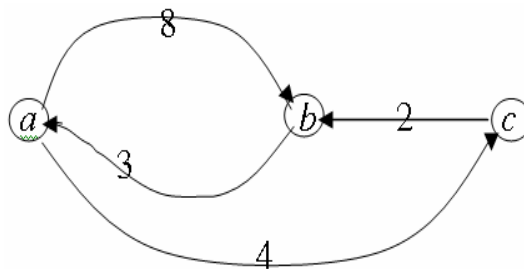
Ta sử dụng ma trận  $D_{n \times n}$  để tính độ dài đường đi ngắn nhất giữa tất cả các cặp đỉnh.

1. Bắt đầu gán  $D := C$  - ma trận trọng số.
2. Thực hiện  $n$  lần lặp trên  $D$ . Sau bước lặp thứ  $k$ ,  $D[i, j]$  chứa độ dài đường đi ngắn nhất từ đỉnh  $i$  đến đỉnh  $j$  mà chỉ đi qua các đỉnh có chỉ số không vượt quá  $k$ . Vậy trong bước lặp thứ  $k$  ta thực hiện theo công thức sau đây:

$$D^{(k)}[i, j] := \min(D^{(k-1)}[i, j], D^{(k-1)}[i, k] + D^{(k-1)}[k, j]),$$

với  $k = 1, 2, \dots, n$ .

Ví dụ 8.5: Giả sử ta có bản đồ giao thông sau đây:



Hình 8.4. Bản đồ giao thông

Các kết quả tính toán:

0	8	4
3	0	$\infty$
$\infty$	2	0

$D_0$

0	8	4
3	0	<u>7</u>
$\infty$	2	0

$D_1$

0	8	4
3	0	7
<u>5</u>	2	0

$D_2$

0	<u>6</u>	4
3	0	7
5	2	0

$D_3$

**Thuật toán 8.5 (Floyd):**

*Dữ liệu:* Ma trận trọng số  $C$  của đồ thị.

*Kết quả:* Ma trận  $D$  cho biết khoảng cách của tất cả các cặp đỉnh.

```

1 BEGIN
2   for  $i := 1$  to  $n$  do
3     for  $j := 1$  to  $n$  do
4       begin  $D[i,j] := C[i,j]$  ;  $TRUOC[i,j] := 0$  end ;
5     for  $k := 1$  to  $n$  do
6       for  $i := 1$  to  $n$  do
7         for  $j := 1$  to  $n$  do
8           if  $D[i,k] + D[k,j] < D[i,j]$  then
9             begin
10               $D[i,j] := D[i,k] + D[k,j]$  ;
11               $TRUOC[i,j] := k$ 
12            end ;
13   END .

```

Nếu  $TRUOC[i,j] = 0$  thì đường đi ngắn nhất từ đỉnh  $i$  đến đỉnh  $j$  chính là cạnh  $(i, j)$ .

Để in ra các đỉnh trung gian trên đường đi ngắn nhất từ đỉnh  $i$  đến đỉnh  $j$  ta dùng thủ tục đệ quy sau đây:

```

1 procedure Duong_di(  $i, j$  ) ;
2 begin
3    $k := TRUOC[i,j]$  ;
4   if  $k = 0$  then Exit ;
5   Duong_di(  $i, k$  ) ;
6   write(  $k$  ) ;
7   Duong_di(  $k, j$  ) ;
8 end ;

```

Chẳng hạn, ma trận TRUOC của ví dụ trên là:

0	3	0
0	0	1
2	0	0

Để xác định đường đi ngắn nhất từ đỉnh 1 đến đỉnh 2 ta lấy  $k = \text{TRUOC}[1,2] = 3$ .  
 Vậy đường đi ngắn nhất là:  $\langle 1, 3, 2 \rangle$ .

### 8.6. Tâm của đồ thị

Giả sử  $G = (V, E)$  là một đồ thị có trọng số không âm trên các cạnh.  
 Ký hiệu:  $d(x,y)$  là khoảng cách giữa đỉnh  $x$  và đỉnh  $y$  trong đồ thị  $G$ .

Đại lượng  $d(a) = \max \{ d(x,a) \mid x \in V \}$  được gọi là *độ lệch* của đỉnh  $a$  trong đồ thị  $G$ .

Các khái niệm bán kính, tâm và đường kính của một đồ thị được định nghĩa như sau:

**Định nghĩa 8.6:**

1. *Bán kính*  $R$  của đồ thị  $G$  là độ lệch bé nhất trên các đỉnh:

$$R = \min \{ d(a) \mid a \in V \}.$$

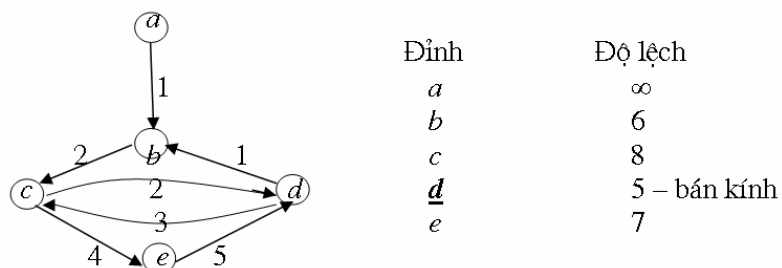
2. *Tâm* của đồ thị  $G$  là đỉnh  $a$  có độ lệch bé nhất:

$$\forall x \in V, \quad d(a) \leq d(x).$$

3. *Đường kính* của đồ thị là khoảng cách dài nhất giữa các cặp đỉnh trong đồ thị:

$$d = \max \{ d(x,y) \mid x, y \in V \}.$$

Ví dụ 8.7: Cho đồ thị  $G$  như sau:



Hình 8.5. Đồ thị có trọng số



Vậy tâm của đồ thị là đỉnh  $d$ .

Ý nghĩa của tâm đồ thị: Dùng để xác định thủ đô của một nước, nút giao thông quan trọng trong một thành phố, vị trí đặt máy chủ trong một mạng máy tính ...

**Thuật toán 8.6** (Tìm tâm của đồ thị):

- 1) Dùng thuật toán Floyd để tính ma trận  $D$  các khoảng cách của các cặp đỉnh.
- 2) Tìm giá trị lớn nhất trên mỗi cột, cho ta độ lệch của đỉnh tương ứng.
- 3) Tìm đỉnh với độ lệch bé nhất, đó chính là tâm của đồ thị.

Ma trận khoảng cách  $D$  của ví dụ trên là:

$a$	$b$	$c$	$d$	$e$
0	1	3	5	7
$\infty$	0	2	4	6
$\infty$	3	0	2	4
$\infty$	1	3	0	7
$\infty$	6	8	5	0
$\infty$	6	8	<b>5</b>	7

Tâm:  $d$   
Bán kính: 5  
Đường kính:  $\infty$

Nếu vẽ một vòng tròn có tâm và bán kính như định nghĩa thì tất cả các đỉnh của đồ thị sẽ nằm trong vòng tròn này.