Quy hoạch động

Trần Vĩnh Đức

HUST

Ngày 7 tháng 9 năm 2019

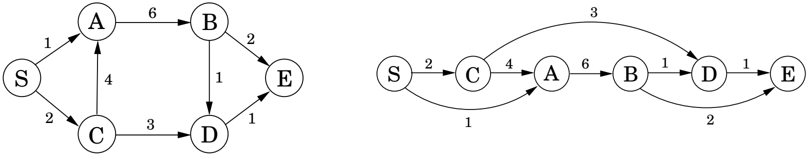
Tài liệu tham khảo

- S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*, July 18, 2006.

Nội dung

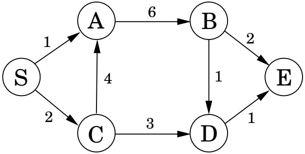
Đồ thị phi chu trình (DAG): Nhắc lại

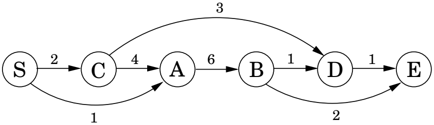
Trong đồ thị phi chu trình, ta có thể sắp xếp thứ tự các đỉnh sao cho nó chỉ có cung đi đi từ trái sang phải.



Hình: Đồ thị phi chu trình G và biểu diễn dạng tuyến tính của nó.

Đường đi ngắn nhất trên DAG



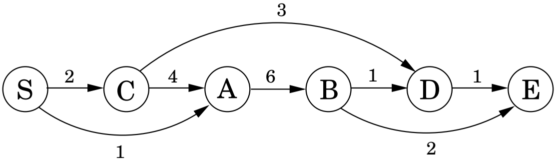


Hình: Đồ thị phi chu trình G và biểu diễn dạng tuyến tính của nó.

- Xét nút D của đồ thị, cách duy nhất để đi từ S đến D là phải qua B hoặc C.
- Vậy, để tìm đường đi ngắn nhất từ S tới D ta chỉ phải so sánh hai đường:

$$\text{dist}(D) = \min\{\text{dist}(B) + 1, \text{dist}(C) + 3\}$$

Thuật toán tìm đường đi ngắn nhất cho DAG



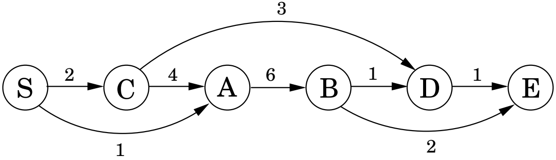
Thuật toán

Khởi tạo mọi giá trị $\mathrm{dist}(.)$ bằng ∞

$\mathrm{dist}(s) = 0$

for each $v \in V \setminus \{s\}$, theo thứ tự tuyến tính:

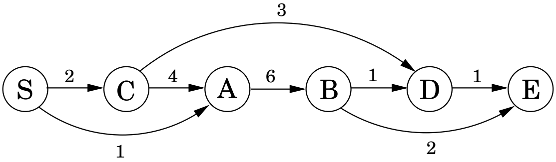
$$\mathrm{dist}(v) = \min_{(u,v) \in E}\{\mathrm{dist}(u) + \ell(u, v)\}$$



Bài tập

Làm thế nào để tìm đường đi dài nhất trong DAG?

Ý tưởng quy hoạch động



Hình: Để giải bài toán D ta cần giải bài toán con C và B.

- ***Quy hoạch động*** là kỹ thuật giải bài toán bằng cách xác định một tập các bài toán con và giải từng bài toán con một, nhỏ nhất trước,
- dùng câu trả lời của bài toán nhỏ để hình dung ra đáp án của bài toán lớn hơn,
- cho tới khi toàn bộ các bài toán được giải.

Nội dung

Bài toán dãy con tăng dài nhất

Cho dãy số $a_1, a_2, \ldots, a_n$. Một dãy con là một tập con các số lấy theo thứ tự, nó có dạng

$$a_{i_1}, a_{i_2}, \ldots, a_{i_k}$$

ở đó $1 \leq i_1 < i_2 < \cdots < i_k \leq n$, và dãy tăng là dãy mà các phần tử tăng dần. Nhiệm vụ của bạn là tìm dãy tăng có số phần tử nhiều nhất.

Ví dụ

Dãy con dài nhất của dãy $5, 2, 8, 6, 3, 6, 9, 7$ là:

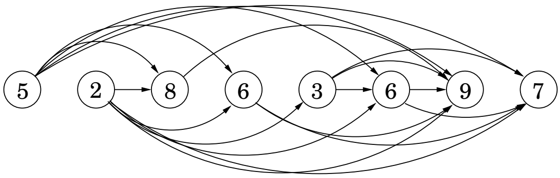
$$5 \quad 2 \quad 8 \quad 6 \quad 3 \quad 6 \quad 9 \quad 7$$

Bài tập

Hãy tìm dãy con tăng dài nhất của dãy

$$5,\ 3,\ 8,\ 6,\ 2,\ 6,\ 9,\ 7.$$

DAG của dãy tăng



Hình: DAG $G = (V, E)$ của dãy $5, 2, 8, 6, 3, 6, 9, 7$.

Ta xây dựng DAG $G = (V, E)$ cho dãy $a_1,\ a_2,\ \ldots,\ a_n$ như sau:

- $V = \{a_1, a_2, \ldots, a_n\}$, và
- có cung $(a_i, a_j) \in E$ nếu $i < j$ và $a_i < a_j$.

Bài toán tìm dãy tăng dài nhất được đưa về bài toán tìm đường đi dài nhất trên DAG.

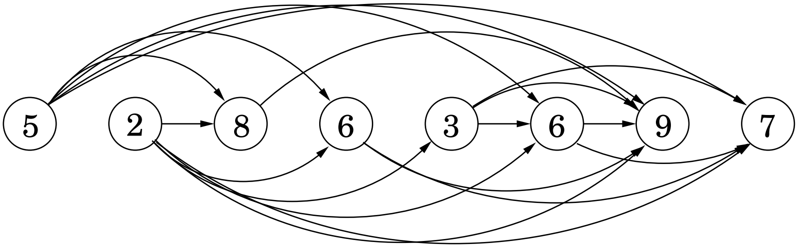
Tìm đường đi dài nhất trên DAG

```
for j = 1, 2, ..., n:
    L(j) = 1 + max{L(i) : (i, j) ∈ E}
    return max_j L(j)
```

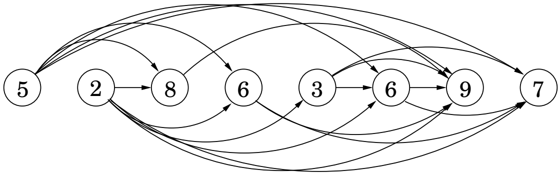
- $L(j)$ là độ dài của đường đi dài nhất kết thúc tại j.

Bài tập

Hãy tìm đường đi dài nhất trên DAG sau:

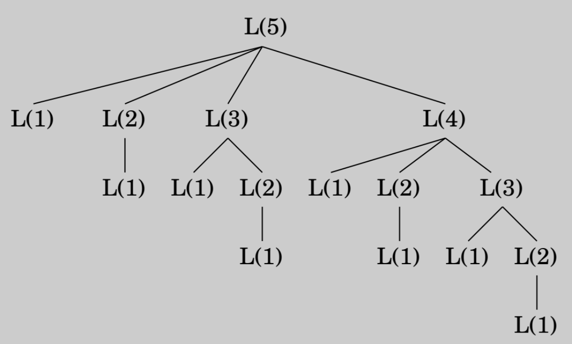


Tìm đường đi dài nhất



- Làm thế nào tìm được đường đi dài nhất từ các L-giá trị?
- Ta quản lý các cạnh trên đường đi bởi con trỏ ngược $\texttt{prev}(j)$ giống như trong tìm đường đi ngắn nhất.
- Dãy tối ưu có thể tìm được theo con trỏ ngược này.

Ta có nên dùng đệ quy?



Có nên dùng đệ quy để tính công thức sau?

$$L(j) = 1 + \max\{L(i) : (i, j) \in E\}$$

Nội dung

Khoảng cách soạn thảo

- Khi chương trình kiểm tra chính tả bắt gặp lỗi chính tả, nó sẽ tìm trong từ điển một từ gần với từ này nhất.
- Ký hiệu thích hợp cho khái niệm gần trong trường hợp này là gì?

Ví dụ

Khoảng cách giữa hai xâu `SNOWY` và `SUNNY` là gì?

```
S - N O W Y        - S N O W - Y
S U N N - Y        S U N - - N Y
```

Chi phí: 3 | Chi phí: 5

Khoảng cách soạn thảo

Định nghĩa

Khoảng cách soạn thảo của hai xâu x và y là số tối thiểu phép toán soạn thảo (xóa, chèn, và thay thế) để biến đổi xâu x thành xâu y.

Ví dụ

Biến đổi xâu `SNOWY` thành xâu `SUNNY`.

```
S - N O W Y
S U N N - Y
```

- chèn `U`,
- thay thế `O` $\rightarrow$ `N`, và
- xóa `W`.

Lời giải quy hoạch động

Câu hỏi

Bài toán con là gì?

- Để tìm khoảng cách soạn thảo giữa hai xâu $x[1 \ldots m]$ và $y[1 \ldots n]$,
- ta xem xét khoảng cách soạn thảo giữa hai khúc đầu

$$x[1 \ldots i] \text{ và } y[1 \ldots j].$$

- Ta gọi đây là bài toán con $E(i, j)$.
- Ta cần tính $E(m, n)$.

Ví dụ

$$\boxed{\text{E X P O N E N}}\ \text{T I A L}$$

$$\boxed{\text{P O L Y N}}\ \text{O M I A L}$$

Hình: Bài toán con $E(7,5)$

Để giải bài toán $E(8,6)$, ta xem xét các khả năng gióng hàng của hai ký tự ở vị trí $x[8]$ và $y[6]$:

$$\frac{x[8]}{-} \quad \text{hoặc} \quad \frac{-}{y[6]} \quad \text{hoặc} \quad \frac{x[8]}{y[6]}$$

Cụ thể

$$\frac{\mathtt{T}}{-} \quad \text{hoặc} \quad \frac{-}{\mathtt{0}} \quad \text{hoặc} \quad \frac{\mathtt{T}}{\mathtt{0}}$$

Bài toán con

- Làm thế nào để tính $E(i, j)$ từ các bài toán con?
- Làm thế nào để gióng hàng $x[1 \dots i]$ và $y[1 \dots j]$?
- Việc gióng hàng ở cột phải nhất có thể chia làm ba trường hợp:

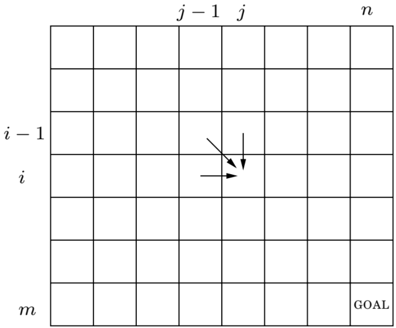
$$\frac{x[i]}{-} \quad \text{hoặc} \quad \frac{-}{y[j]} \quad \text{hoặc} \quad \frac{x[i]}{y[j]}$$

Ta có công thức

$$E(i, j) = \min \begin{cases} E(i-1, j) + 1, \\ E(i, j-1) + 1, \\ E(i-1, j-1) + \texttt{diff}(i, j) \end{cases}$$

ở đó $\texttt{diff}(i, j) = 0$ nếu $x[i] = y[j]$ và 1 nếu ngược lại.

Đưa về bài toán nhỏ hơn



$$E(i,j) = \min \begin{cases} E(i-1,j)+1, \\ E(i,j-1)+1, \\ E(i-1,j-1)+\texttt{diff}(i,j) \end{cases}$$

ở đó $\texttt{diff}(i,j) = 0$ nếu $x[i] = y[j]$ và 1 nếu ngược lại.

Thuật toán quy hoạch động

$$
\begin{aligned}
&\texttt{for}\ \ i = 0, 1, \ldots, m:\\
&\qquad E(i, 0) = i\\
&\texttt{for}\ \ j = 1, 2, \ldots, n:\\
&\qquad E(0, j) = j\\
&\texttt{for}\ \ i = 1, 2, \ldots, m:\\
&\qquad \texttt{for}\ \ j = 1, 2, \ldots, n:\\
&\qquad\qquad E(i, j) = \min \begin{cases} E(i-1, j) + 1, \\ E(i, j-1) + 1, \\ E(i-1, j-1) + \texttt{diff}(i, j) \end{cases}
\end{aligned}
$$

Bài tập

Hãy đánh giá độ phức tạp của thuật toán.

$$
\begin{aligned}
&\texttt{for}\ \ i = 0, 1, \ldots, m:\\
&\qquad E(i, 0) = i\\
&\texttt{for}\ \ j = 1, 2, \ldots, n:\\
&\qquad E(0, j) = j\\
&\texttt{for}\ \ i = 1, 2, \ldots, m:\\
&\qquad \texttt{for}\ \ j = 1, 2, \ldots, n:\\
&\qquad\qquad E(i, j) = \min \begin{cases} E(i-1, j) + 1, \\ E(i, j-1) + 1, \\ E(i-1, j-1) + \texttt{diff}(i, j) \end{cases}
\end{aligned}
$$

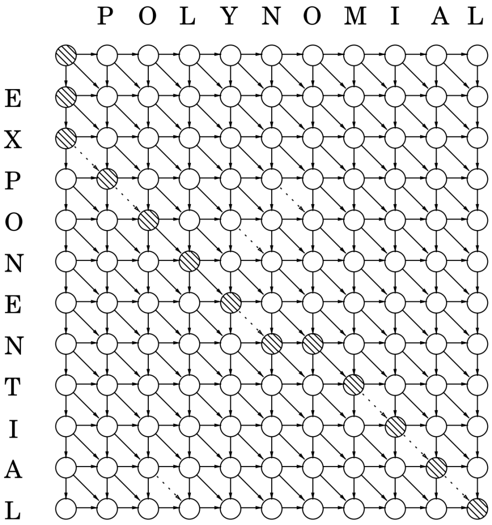
Ví dụ

		P	O	L	Y	N	O	M	I	A	L
	0	1	2	3	4	5	6	7	8	9	10
E	1	1	2	3	4	5	6	7	8	9	10
X	2	2	2	3	4	5	6	7	8	9	10
P	3	2	3	3	4	5	6	7	8	9	10
O	4	3	2	3	4	5	5	6	7	8	9
N	5	4	3	3	4	4	5	6	7	8	9
E	6	5	4	4	4	5	5	6	7	8	9
N	7	6	5	5	5	4	5	6	7	8	9
T	8	7	6	6	6	5	5	6	7	8	9
I	9	8	7	7	7	6	6	6	6	7	8
A	10	9	8	8	8	7	7	7	7	6	7
L	11	10	9	8	9	8	8	8	8	7	6

$$E(i,j) = \min\{1+E(i-1,j), 1+E(i,j-1), \texttt{diff}(i,j)+E(i-1,j-1)\}$$

ở đó $\texttt{diff}(i,j) = 0$ nếu $x[i] = y[j]$ và 1 nếu ngược lại.

DAG của bài toán & đường đi độ dài 6



DAG của bài toán

Xây dựng đồ thị $G = (V, E)$ với

- tập đỉnh ứng với các vị trí (i, j) trong bảng,
- các cung

$$(i{-}1, j) \to (i, j), \quad (i, j{-}1) \to (i, j), \quad (i{-}1, j{-}1) \to (i, j)$$

  đều có trọng số 1, ngoại trừ cung

$$\{(i-1, j-1) \to (i, j) : x[i] = y[j]\} \quad \text{có trọng số } 0.$$

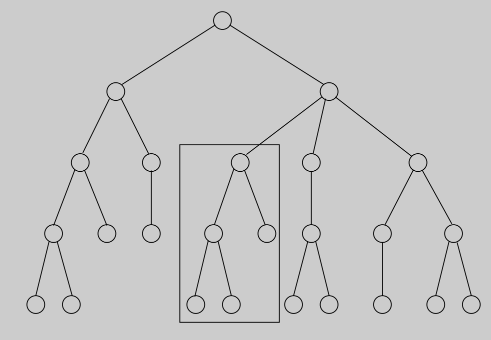
Ta cần tìm đường đi ngắn nhất từ đỉnh $s = (0, 0)$ tới $t = (m, n)$.

Trên đường đi này: sang phải (thêm), đi xuống (xóa), đi chéo (thay thế).

Số lượng các bài toán con

- Đầu vào $x_1, x_2, \ldots, x_n$ và một bài toán con là $x_1, x_2, \ldots, x_i$. Số bài toán con là tuyến tính.
- Đầu vào $x_1, x_2, \ldots, x_n$ và $y_1, y_2, \ldots, y_m$. bài toán con là $x_1, x_2, \ldots, x_i$ và $y_1, y_2, \ldots, y_j$. Số bài toán con là $O(mn)$.
- Đầu vào $x_1, x_2, \ldots, x_n$ và bài toán con là $x_i, x_{i+1}, \ldots, x_j$. Số bài toán con là $O(n^2)$.

Số lượng các bài toán con (2)



- Đầu vào là một cây có gốc. Một bài toán con là một cây con có gốc.
- Nếu cây có n nút thì có thể có bao nhiêu bài toán con?

Nội dung

Bài toán cái túi

- Trong một vụ cướp, tên trộm tìm thấy nhiều chiến lợi phẩm hơn anh ta dự kiến và phải quyết định xem lấy những gì.
- Chiếc túi của anh ấy (hay "knapsack") chứa được tối đa là W pao. Có n vật để chọn có trọng lượng $w_1, \ldots, w_n$ và có giá trị (theo đô la) là $v_1, \ldots, v_n$.
- Hắn nên lấy những mặt hàng nào để được lợi nhất?

Ví dụ

Nên lấy đồ vật nào nếu mỗi đồ vật có số lượng vô hạn? Nếu mỗi đồ vật chỉ có một?

Đồ vật	Trọng lượng	Giá trị ($)
1	6	30
2	3	14
3	4	16
4	2	9

Các đồ vật có thể lấy nhiều lần

Câu hỏi
Bài toán con là gì? bài toán với trọng lượng túi nhỏ hơn $w \leq W$ hay với số đồ vật ít hơn. Ví dụ $1, \ldots, j$ với $j \leq n$.

Đưa về bài toán con với trọng lượng túi nhỏ hơn
Đặt

$$K(w) = \text{ giá trị lớn nhất chọn được khi trọng lượng túi là } w.$$

Khi đó $K(w) = K(w - w_i) + v_i$ với đồ vật i nào đó.

Do không biết là đồ vật nào được thêm vào, ta sẽ thử mọi khả năng:

$$K(w) = \max_{i: w_i \leq w} \{K(w - w_i) + v_i\}.$$

Thuật toán quy hoạch động

$$
\begin{aligned}
&K(0) = 0 \\
&\texttt{for } w = 1 \texttt{ to } W: \\
&\qquad K(w) = \max\{K(w - w_i) + v_i : w_i \leq w\} \\
&\texttt{return } K(W)
\end{aligned}
$$

Câu hỏi

Độ phức tạp của thuật toán này là gì?

Mỗi đồ vật chỉ lấy một lần

Câu hỏi

Bài toán con là gì?

Giá trị $K(w - w_n)$ rất lớn cũng không giúp gì vì ta không biết đồ vật n đã có trong lời giải con chưa.

$K(w, j) =$ giá trị lớn nhất chọn được khi trọng lượng túi là w và các đồ vật là $1, 2, \ldots, j$.

Đưa về bài toán nhỏ hơn

$$K(w, j) = \text{giá trị lớn nhất chọn được khi trọng lượng túi là } w$$
$$\text{và các đồ vật là } 1, 2, \ldots, j.$$

Để biểu diễn $K(w, j)$ về các bài toán nhỏ hơn, ta xét hai trường hợp:

- đồ vật j cần cho giá trị tối ưu,
- hoặc không cần:

$$K(w, j) = \max\{K(w - w_j, j - 1) + v_j,\ K(w, j - 1)\}.$$

Thuật toán quy hoạch động

```
Khởi tạo mọi K(0, j) = 0 và mọi K(w, 0) = 0
for j = 1 to n:
    for w = 1 to W:
        if w_j > w:  K(w, j) = K(w, j–1)
        else:  K(w, j) = max{K(w–w_j, j–1) + v_j,  K(w, j–1)}
return K(W, n)
```

Câu hỏi

Độ phức tạp của thuật toán này là gì?

Nội dung

Ví dụ nhân nhiều ma trận

- Giả sử ta muốn nhân 4 ma trận

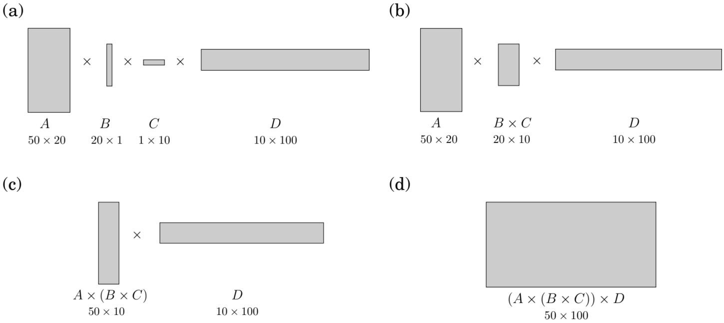
$$A \times B \times C \times D$$

với số chiều là

$$50 \times 20, \quad 20 \times 1, \quad 1 \times 10, \quad \text{và} \quad 10 \times 100.$$

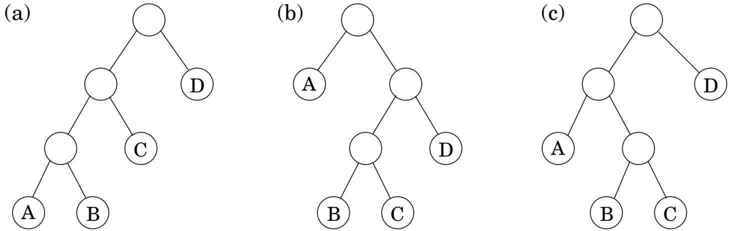
- Do tính chất kết hợp của phép nhân ma trận, ta có thể nhân theo nhiều cách.

Cách đặt ngoặc	Tính toán	Chi phí
$A \times ((B \times C) \times D)$	$20 \cdot 1 \cdot 10 + 20 \cdot 10 \cdot 100 + 50 \cdot 20 \cdot 100$	120,200
$(A \times (B \times C)) \times D$	$20 \cdot 1 \cdot 10 + 50 \cdot 20 \cdot 10 + 50 \cdot 10 \cdot 100$	$60,200$
$(A \times B) \times (C \times D)$	$50 \cdot 20 \cdot 1 + 1 \cdot 10 \cdot 100 + 50 \cdot 1 \cdot 100$	$7,000$



Hình: $A \times B \times C \times D = (A \times (B \times C)) \times D$.

Mỗi phép nhân ứng với một cây nhị phân



Bài toán

- Ta cần tính
$$A_1 \times A_2 \times \cdots \times A_n$$
trong đó các A_i có số chiều tương ứng là
$$m_0 \times m_1,\ m_1 \times m_2,\ \ldots\ m_{n-1} \times m_n.$$
- Hãy xác định thứ tự tính toán tối ưu.

Bài toán con

Đặt

$$C(i,j) = \text{ chi phí tối thiểu để nhân } A_i \times A_{i+1} \times \cdots \times A_j.$$

Cách tính tối ưu sẽ là một cách đặt ngoặc:

$$(A_i \times A_{i+1} \times \cdots \times A_k) \times (A_{k+1} \times A_{k+2} \cdots \times A_j)$$

với chi phí tối thiểu là

$$C(i,j) = C(i,k) + C(k+1,j) + m_{i-1} \cdot m_k \cdot m_j.$$

Ta không biết vị trí k ở đâu nên phải thử mọi khả năng:

$$C(i,j) = \min_{i \le k < j} \{ C(i,k) + C(k+1,j) + m_{i-1} \cdot m_k \cdot m_j \}.$$

Thuật toán quy hoạch động

for $i = 1$ to n: $C(i, i) = 0$
for $s = 1$ to $n-1$:
    for $i = 1$ to $n-s$:
        $j = i + s$

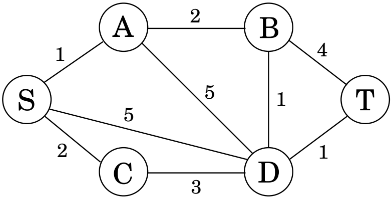
$C(i, j) = \min\{C(i, k) + C(k+1, j) + m_{i-1} \cdot m_k \cdot m_j : i \leq k < j\}$
return $C(1, n)$

Câu hỏi

Độ phức tạp của thuật toán này là gì?

Nội dung

Đường đi tin cậy ngắn nhất



Hình: Ta muốn tìm đường đi từ s tới t vừa ngắn vừa ít cạnh.

- Trong mạng truyền thông, dù độ dài cạnh có liên quan đến độ trễ, ta có một số cân nhắc liên quan đến việc chọn đường đi.
- Mỗi cạnh thêm vào đường đi có thể gây ra nguy cơ mất gói tin.
- Ta muốn tránh đường đi có quá nhiều cạnh.

Bài toán đường đi tin cậy ngắn nhất

- Cho đồ thị trọng số G, hai đỉnh s và t, và một số nguyên dương k,
- ta muốn tìm đường đi ngắn nhất từ s tới t qua nhiều nhất k cạnh.

$$\texttt{dist}(v, i) = \text{ độ dài đường đi ngắn nhất từ } s \text{ đến } v \text{ dùng } i \text{ cạnh}$$

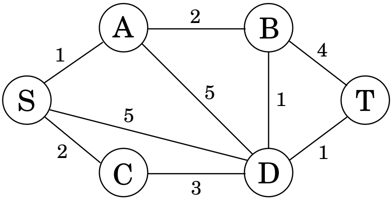
Ta có

$$\texttt{dist}(v, i) = \min_{(u,v) \in E} \{\texttt{dist}(u, i-1) + \ell(u, v)\}$$

Bài toán tìm đường ngắn nhất giữa mọi cặp đỉnh

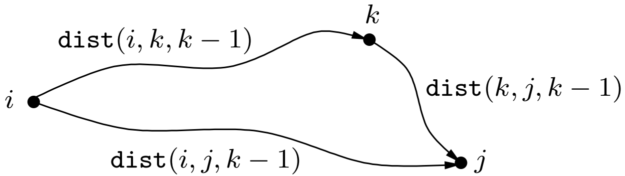
Bài toán

- Các đỉnh của $V = \{1, 2, \ldots, n\}$.
- Ta cần tìm đường đi ngắn nhất giữa mọi cặp đỉnh.



Bài toán con

$$\texttt{dist}(i, j, k) = \text{ độ dài đường đi ngắn nhất từ } i \text{ tới } j$$
$$\text{chỉ qua các đỉnh trung gian } \{1, 2, \ldots, k\}$$



Đường đi i đến j qua đỉnh k sẽ ngắn hơn là không qua đỉnh k nếu và chỉ nếu:

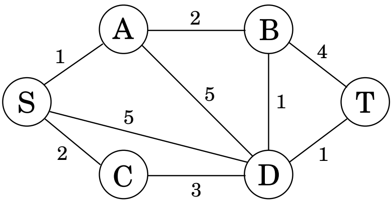
$$\texttt{dist}(i, k, k-1) + \texttt{dist}(k, j, k-1) < \texttt{dist}(i, j, k-1),$$

Thuật toán quy hoạch động

```
for i = 1 to n:
    for j = 1 to n:
        dist(i, j, 0) = ∞
for all (i, j) ∈ E:
    dist(i, j, 0) = ℓ(i, j)
for k = 1 to n:
    for i = 1 to n:
        for j = 1 to n:
```

$$\texttt{dist}(i,j,k) = \min \begin{cases} \texttt{dist}(i,k,k{-}1) + \texttt{dist}(k,j,k{-}1), \\ \texttt{dist}(i,j,k{-}1) \end{cases}$$

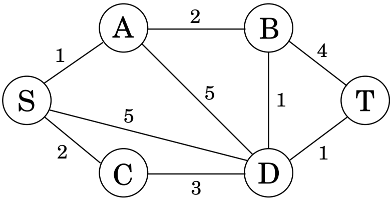
Bài toán người bán hàng du lịch



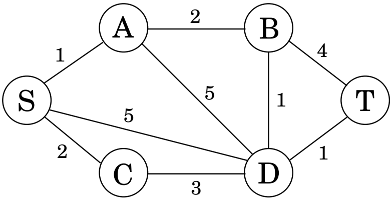
- Một người bán hàng thích du lịch đang sẵn sàng cho một chuyến đi bán hàng.
- Bắt đầu từ nhà, với chiếc vali trong tay, anh sẽ đi một hành trình trong đó mỗi thành phố cần đến sẽ được thăm đúng một lần trước khi quay về nhà.
- Cho khoảng cách d_{ij} giữa mỗi cặp thành phố i và j, anh ta nên thăm các thành phố theo thứ tự nào để tối ưu tổng khoảng cách?

Bài tập

Người bán hàng nên thăm các thành phố theo thứ tự nào để tối ưu tổng khoảng cách?



Bài toán con



Với mỗi tập con các thành phố $S = \{1, 2, \ldots, n\}$ có chứa 1, và $j \in S$, ta ký hiệu $C(S, j)$ là độ dài của đường đi ngắn nhất thăm mỗi đỉnh của S đúng một lần, bắt đầu từ 1 và kết thúc tại j.

Đưa về bài toán nhỏ hơn

- Nếu ta bắt đầu tại 1 và kết thúc tại j, ta nên chọn đến đỉnh nào trước khi đến j?
- Đỉnh này sẽ là một đỉnh $i \in S$ thỏa mãn: độ dài đường đi ngắn nhất từ 1 đến i cộng với d_{ij} là nhỏ nhất.

$$C(S,j) = \min_{i \in S: i \neq j} C(S - \{j\}, i) + d_{ij}$$

- Khi $|S| > 1$, ta ký hiệu $C(S,1) = \infty$ vì đường đi không thể vừa bắt đầu và kết thúc ở 1.

Thuật toán quy hoạch động

```
C({1}, 1) = 0
for s = 2 to n:
  for all tập con S ⊆ {1, 2, ..., n} thỏa mãn |S| = s và
1 ∈ S:
     C(S, 1) = ∞
     for all j ∈ S, j ≠ 1:
          C(S, j) = min{C(S − {j}, i) + d_ij : i ∈ S, i ≠ j}
return min_j C({1, ..., n}, j) + d_j1
```

Có nhiều nhất $2^n \cdot n$ bài toán con, và mỗi bài mất thời gian tuyến tính để giải. Độ phức tạp là $O(n^2 \cdot 2^n)$.

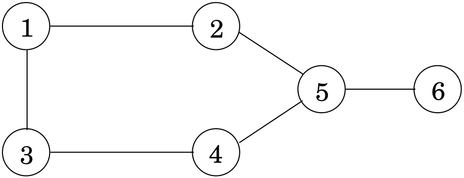
Nội dung

- Đường đi ngắn nhất trên DAG
- Dãy con tăng dài nhất
- Khoảng cách soạn thảo
- Bài toán cái túi
- Nhân nhiều ma trận
- Đường đi ngắn nhất
- Tập độc lập trên cây

Tập độc lập

Định nghĩa

Một tập con các đỉnh $S \subseteq V$ là một ***tập độc lập*** của đồ thị $G = (V, E)$ nếu không có cạnh giữa chúng.



- $\{1, 5\}$ là một tập độc lập, nhưng $\{1, 4, 5\}$ không phải.
- Tập độc lập lớn nhất là $\{2, 3, 6\}$.

Tập độc lập trên cây

- Bài toán tìm tập độc lập lớn nhất được nhiều người tin rằng không có thuật toán hiệu quả để giải nó.
- Nhưng khi đồ thị là một cây thì bài toán có thể giải trong thời gian tuyến tính.

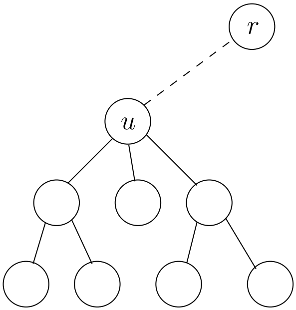
Bài toán con

Lấy một nút r bất kỳ làm gốc của cây. Mỗi nút u bây giờ sẽ xác định một cây con gốc u. Ta xét bài toán con:

$$I(u) = \text{ kích thước tập độc lập lớn nhất của cây con gốc } u.$$

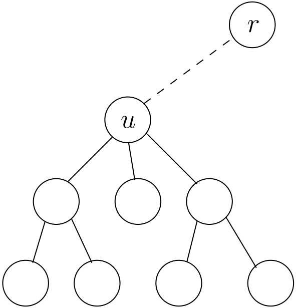
Mục đích của ta là tìm $I(r)$.

Đưa về bài toán nhỏ hơn



- Giả sử ta đã biết tập độc lập lớn nhất của mọi cây con bắt đầu từ u; tức là ta đã biết $I(w)$ cho mọi con cháu w của u.
- Làm thế nào để tính $I(u)$?
- Tách thành hai trường hợp: tập độc lập hoặc chứa u hoặc nó không chứa u.

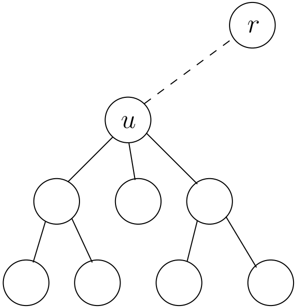
Đưa về bài toán nhỏ hơn



Hai trường hợp: tập độc lập hoặc chứa u hoặc nó không chứa u.

$$I(u) = \max\left\{1 + \sum_{\text{các cháu } w \text{ của } u} I(w), \sum_{\text{các con } w \text{ của } u} I(w)\right\}$$

Độ phức tạp tính toán



Bài tập

Số cây con chính là số đỉnh. Liệu ta có thể cài đặt thuật toán chạy trong

$$O(|V| + |E|)?$$