

# Bài tập Tắc nghẽn

## Một số thuật ngữ:

- **Max:** Yêu cầu ban đầu (ma trận  $m \times n$ , với  $m$  là số dòng - ứng với số lượng tiến trình,  $n$  là cột - ứng với số lượng tài nguyên). Trong một số tài liệu, người ta thường dùng từ **Request** thay cho **Max**.
- **Allocation:** Đã cấp phát (ma trận  $m \times n$ )
- **Available:** Tài nguyên còn lại (ma trận  $1 \times n$ )
- **Need:** Nhu cầu còn lại (ma trận  $m \times n$ , xác định như sau:  $Need[i,j] = Max[i,j] - Allocation[i,j]$ )
- **Số tài nguyên từng loại:**  $Allocation[j] + Available[j]$

## Bài 1.

Một hệ thống có 3 loại tài nguyên (A, B, C) và 5 tiến trình (P0, P1, P2, P3, P4) kèm theo các thông số được mô tả trong bảng sau.

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	3	0	1	10	7	4	6	2	2
P1	3	2	1	8	5	3			
P2	2	1	3	6	3	4			
P3	0	3	0	9	6	3			
P4	1	1	2	7	4	5			

Tiến trình P1 yêu cầu tài nguyên là (2, 0, 1). Sử dụng giải thuật Banker, cho biết có thể thực hiện yêu cầu cấp phát tài nguyên này hay không?

## GIẢI

**Bước 1:** Kiểm tra  $Request \leq Available$

$$2 \ 0 \ 1 \leq 6 \ 2 \ 2 \quad \text{True}$$

Yêu cầu là hợp lệ.

Thử kiểm tra việc cấp phát có an toàn không

**Bước 2:**  $Work = Available - Request = 6 \ 2 \ 2 - 2 \ 0 \ 1 = 4 \ 2 \ 1$

Cập nhật Allocation cho P1 =  $3 \ 2 \ 1 + 2 \ 0 \ 1 = 5 \ 2 \ 2$

**Bước 3:** Tính  $Need = Max - Allocation$

$$P0: 10 \ 7 \ 4 - 3 \ 0 \ 1 = 7 \ 7 \ 3$$

$$P1: 8 \ 5 \ 3 - 5 \ 2 \ 2 = 3 \ 3 \ 1$$

$$P2: 6 \ 3 \ 4 - 2 \ 1 \ 3 = 4 \ 2 \ 1$$

$$P3: 9 \ 6 \ 3 - 0 \ 3 \ 0 = 9 \ 3 \ 3$$

$$P4: 7 \ 4 \ 5 - 1 \ 1 \ 2 = 6 \ 3 \ 3$$

**Bước 4:** Xác định  $Need(i) \leq Work$

$$\text{Với } P0: 7 \ 7 \ 3 \leq 4 \ 2 \ 1 \rightarrow \text{False}$$

$$\text{Với } P1: 3 \ 3 \ 1 \leq 4 \ 2 \ 1 \rightarrow \text{False}$$

$$\text{Với } P2: 4 \ 2 \ 1 \leq 4 \ 2 \ 1 \rightarrow \text{True}$$

$\Rightarrow$  Thu hồi tài nguyên  $Work = Work + Allocation(P2) = (4, 2, 1) + (2, 1, 3) = (6, 3, 4)$   
 $\Rightarrow$  Xét lại vòng lặp  
 Với P0:  $7\ 7\ 3 \leq (6\ 3\ 4) \rightarrow$  False  
 Với P1:  $3\ 3\ 1 \leq (6\ 3\ 4) \rightarrow$  True  
 $\Rightarrow$  Thu hồi tài nguyên  $Work = Work + Allocation(P1) = (6, 3, 4) + (5, 2, 2) = (11, 5, 6)$   
 $\Rightarrow$  Xét lại vòng lặp  
 Với P0:  $7\ 7\ 3 \leq (11\ 5\ 6) \rightarrow$  False  
 Với P3:  $9\ 3\ 3 \leq (11\ 5\ 6) \rightarrow$  True  
 $\Rightarrow$  Thu hồi tài nguyên  $Work = Work + Allocation(P3) = 11\ 5\ 6 + 0\ 3\ 0 = 11\ 8\ 6$   
 $\Rightarrow$  Xét lại vòng lặp  
 Với P0:  $7\ 7\ 3 \leq (11\ 8\ 6) \rightarrow$  True  
 $\Rightarrow$  Thu hồi tài nguyên  $Work = Work + Allocation(P0) = 11\ 8\ 6 + 3\ 0\ 1 = 14\ 8\ 7$   
 $\Rightarrow$  Xét lại vòng lặp  
 Với P4:  $6\ 3\ 3 \leq (14\ 8\ 7) \rightarrow$  True  
 $\Rightarrow$  Thu hồi tài nguyên  $Work = Work + Allocation(P4) = 14\ 8\ 7 + 1\ 1\ 2 = 15\ 9\ 9$

Tìm thấy chuỗi cấp phát an toàn {P2, P1, P3, P0, P4} nên có thể thực hiện cấp phát tài nguyên cho P1 được.

## **Bài 2.**

Một hệ thống có 3 loại tài nguyên (A, B, C) và 4 tiến trình (P0, P1, P2, P3, P4) kèm theo các thông số được mô tả trong bảng sau.

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	3	0	1	10	7	4	6	2	2
P1	3	2	1	8	5	3			
P2	2	1	3	6	3	4			
P3	0	3	0	9	6	3			
P4	1	1	2	7	4	5			

Tiến trình P1 yêu cầu tài nguyên là (1, 1, 0). Sử dụng giải thuật Banker, cho biết có thể thực hiện yêu cầu cấp phát tài nguyên này hay không?

GIẢI

**Bước 1:** Kiểm tra Request  $\leq$  Available

$$1\ 1\ 0 \leq 6\ 2\ 2 \text{ True}$$

Yêu cầu là hợp lệ

Thử kiểm tra yêu cầu có thể được thực hiện hay không

**Bước 2:**  $Work = Available - Request$

$$6\ 2\ 2 - 1\ 1\ 0 = 5\ 1\ 2$$

Cập nhật lại Allocation cho P1:  $3\ 2\ 1 + 1\ 1\ 0 = 4\ 3\ 1$

**Bước 3:** Tính lại Need = Max - Allocation

$$P0: 10\ 7\ 4 - 3\ 0\ 1 = 7\ 7\ 3$$

$$P1: 8\ 5\ 3 - 4\ 3\ 1 = 4\ 2\ 2$$

P2: 6 3 4 – 2 1 3 = 4 2 1  
P3: 9 6 3 – 0 3 0 = 9 3 3  
P4: 7 4 5 – 1 1 2 = 6 3 3

**Bước 4:** Xác định Need (i) <= Work

Với P0: 7 7 3 <= 5 1 2 -> False

Với P1: 4 2 2 <= 5 1 2 -> False

Với P2: 4 2 1 <= 5 1 2 -> False

Với P3: 9 3 3 <= 5 1 2 -> False

Với P4: 6 3 3 <= 5 1 2 -> False

Không tìm thấy chuỗi cấp phát an toàn, vậy nên không thể thực hiện yêu cầu cấp phát cho P1 được.

**Bài 3.**

Cho hệ thống có 5 tiến trình và 3 loại tài nguyên (A, B, C). Giả sử hệ thống đang ở trạng thái sau:

	Yêu cầu ban đầu (Max)			Đã cấp phát (Allocation)			Tài nguyên rảnh (Available)		
	A	B	C	A	B	C	A	B	C
P1	2	3	2	0	1	2	2	3	3
P2	3	7	5	0	0	1			
P3	3	4	3	2	0	0			
P4	2	2	2	1	1	1			
P5	2	9	0	2	4	0			

- Tính số tài nguyên mỗi loại của hệ thống.
- Hãy tìm một trạng thái an toàn (safe state).
- Nếu tiến trình P3 có yêu cầu thêm tài nguyên (A: 0, B: 0, C: 3), áp dụng giải thuật nhà băng (Banker's Algorithm), xét xem có nên cấp phát cho P3 hay không? Tại sao?

**Bài 4.**

Cho hệ thống có 5 tiến trình và 3 loại tài nguyên (A, B, C). Giả sử hệ thống đang ở trạng thái sau:

	Yêu cầu ban đầu (Max)			Đã cấp phát (Allocation)			Tài nguyên rảnh (Available)		
	A	B	C	A	B	C	A	B	C
P1	3	4	3	2	0	0	2	3	3
P2	3	7	5	0	0	1			
P3	2	3	2	0	1	0			
P4	2	9	0	2	4	0			
P5	2	2	2	1	1	1			

- Tính số tài nguyên mỗi loại của hệ thống.
- Hãy tìm một trạng thái an toàn (safe state).
- Nếu tiến trình P3 có yêu cầu thêm tài nguyên (A: 1, B: 2, C: 0), áp dụng giải thuật nhà băng (Banker's Algorithm), xét xem có nên cấp phát cho P3 hay không? Tại sao?

### **Bài 5.**

Cho hệ thống có 5 tiến trình và 4 loại tài nguyên (A, B, C, D). Giả sử hệ thống đang ở trạng thái sau:

	Yêu cầu ban đầu (Max)				Đã cấp phát (Allocation)				Tài nguyên rảnh (Available)			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	0	0	1	2	0	0	1	1	1	5	2	0
P2	1	7	5	0	1	0	0	0				
P3	2	3	5	6	1	3	5	4				
P4	0	6	5	2	0	6	3	2				
P5	0	9	5	6	0	0	1	4				

- Tính số tài nguyên mỗi loại của hệ thống.
- Tính nhu cầu còn lại (Need) của hệ thống.
- Hãy tìm một trạng thái an toàn (safe state).
- Nếu tiến trình P2 có yêu cầu thêm tài nguyên (A: 0, B: 4, C: 2, D: 0), áp dụng giải thuật nhà băng (Banker's Algorithm), xét xem có nên cấp phát cho P2 hay không? Tại sao?

### **Bài 6.**

Cho hệ thống có 5 tiến trình và 3 loại tài nguyên (A, B, C). Giả sử hệ thống đang ở trạng thái sau:

	Yêu cầu ban đầu (Max)			Đã cấp phát (Allocation)			Tài nguyên rảnh (Available)		
	A	B	C	A	B	C	A	B	C
P1	7	7	3	2	1	0	3	3	2
P2	9	0	2	3	0	2			
P3	4	3	4	0	0	2			
P4	5	2	2	1	1	0			
P5	2	2	2	1	1	1			

- Tính số tài nguyên mỗi loại của hệ thống.
- Hãy tìm một trạng thái an toàn (safe state).
- Nếu tiến trình P3 có yêu cầu thêm tài nguyên (A: 0, B: 0, C: 2), áp dụng giải thuật nhà băng (Banker's Algorithm), xét xem có nên cấp phát cho P3 hay không? Tại sao?

**Bài 7.**

Cho hệ thống có 5 tiến trình và 3 loại tài nguyên (A, B, C). Giả sử hệ thống đang ở trạng thái sau:

	Yêu cầu ban đầu (Max)			Đã cấp phát (Allocation)			Tài nguyên rảnh (Available)		
	A	B	C	A	B	C	A	B	C
P1	7	5	3	2	1	0	3	3	2
P2	9	0	2	3	0	2			
P3	4	3	3	0	0	2			
P4	3	2	2	1	0	0			
P5	2	2	2	1	1	1			

- Tính số tài nguyên mỗi loại của hệ thống.
- Hãy tìm một trạng thái an toàn (safe state).
- Nếu tiến trình P3 có yêu cầu thêm tài nguyên (A: 3, B: 1, C: 0), áp dụng giải thuật nhà băng (Banker's Algorithm), xét xem có nên cấp phát cho P3 hay không? Tại sao?

**Bài 8.**

Xét trạng thái hệ thống:

	Max				Allocation				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P1	0	0	1	2	0	0	1	2	1	5	2	2
P2	1	7	5	0	1	0	0	0				
P3	2	3	5	6	1	3	5	4				
P4	0	6	5	2	0	6	3	2				
P5	0	6	5	6	0	0	1	4				

- Hãy cho biết nội dung của ma trận Need
- Hệ thống có ở trong tình trạng an toàn hay không?
- Với yêu cầu P2(0,4,2,0) thì yêu cầu có được thỏa mãn ngay hay không?

**Bài 9.**

Một hệ thống có 5 tiến trình và 4 loại tài nguyên (A, B, C, D) với tình trạng như sau:

- A có 3 thẻ hiện.
- B có 14 thẻ hiện
- C có 12 thẻ hiện
- D có 12 thẻ hiện

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P <sub>0</sub>	0	0	1	2	0	0	1	2	1	5	2	0
P <sub>1</sub>	1	0	0	0	1	7	5	0				
P <sub>2</sub>	1	3	5	4	2	3	5	6				
P <sub>3</sub>	0	6	3	2	0	6	5	2				
P <sub>4</sub>	0	0	1	4	0	6	5	6				

Dùng giải thuật nhà băng để:

- Chứng minh trạng thái này an toàn.
- Xác định có nên đáp ứng yêu cầu (0, 4, 3, 0) của P<sub>1</sub> ?

### **Bài 10.**

Một hệ thống có 4 tiến trình P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub> và 5 loại tài nguyên R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>, R<sub>4</sub>, R<sub>5</sub>. Trong đó:

- R<sub>1</sub> có 1 thẻ hiện
  - R<sub>2</sub> có 2 thẻ hiện
  - R<sub>3</sub> có 1 thẻ hiện
  - R<sub>4</sub> có 3 thẻ hiện
  - R<sub>5</sub> có 2 thẻ hiện
  - P<sub>1</sub> đang giữ 1 thẻ hiện của R<sub>2</sub> và yêu cầu 1 thẻ hiện của R<sub>1</sub>
  - P<sub>2</sub> đang giữ 1 thẻ hiện của R<sub>1</sub> và yêu cầu 1 thẻ hiện của R<sub>3</sub>
  - P<sub>3</sub> đang giữ 1 thẻ hiện của R<sub>3</sub> và yêu cầu 1 thẻ hiện của R<sub>4</sub>
  - P<sub>4</sub> đang giữ 1 thẻ hiện của R<sub>4</sub> và yêu cầu 1 thẻ hiện của R<sub>2</sub>.
- Hãy vẽ đồ thị cấp phát tài nguyên như mô tả trên.
  - Trạng thái trên có xảy ra deadlock không? Tại sao?

### **Bài 11.**

Trạng thái sau có xảy ra tình trạng deadlock không? Tại sao?

