

---

**TD 02 – Allocation de ressources et interblocages**


---

**Exercice 1.***Algorithme du banquier*

On suppose que le système connaît

- $dispo[i]$  : nombre de ressources  $R_i$  disponibles
- $max[i, j]$  : nombre max de  $R_i$  que  $P_j$  va utiliser
- $alloc[i, j]$  : nombre de  $R_i$  couramment allouées à  $P_j$

La règle est alors la suivante

$$P_j \text{ peut s'exécuter si et seulement si } max[i, j] - alloc[i, j] \leq dispo[i].$$

Un ordre d'exécution  $P_1, P_2, \dots, P_n$  est *sain* si et seulement si on peut exécuter les processus les uns après les autres dans cette ordre.

L'état d'un système est *sain* si et seulement si il existe un ordre d'exécution sain.

(Rappel : l'allocateur reçoit des demandes, et alloue (ou non) les ressources.)

1. Donner l'algorithme (du banquier) de l'allocateur en pseudo-code.
2. Les états suivants sont-ils sains ?

(a) 5 processus ( $A, B, C, D, E$ )

4 ressources ( $R_1, R_2, R_3, R_4$ ) en quantités respectives (6, 4, 3, 2).

Processus	Ressources attribuées				Ressources demandées			
	$R_1$	$R_2$	$R_3$	$R_4$	$R_1$	$R_2$	$R_3$	$R_4$
A	3	0	1	1	1	1	0	0
B	0	1	0	0	0	1	1	2
C	1	1	1	0	3	1	0	0
D	1	1	0	1	0	0	1	0
E	0	0	0	0	2	1	1	0

(cet exemple est tiré de la page Wikipédia sur l'algorithme du banquier)

(b) 5 processus ( $A, B, C, D, E$ )

4 ressources ( $R_1, R_2, R_3, R_4$ ) en quantités respectives (12, 7, 4, 5).

Processus	Ressources attribuées				Ressources demandées			
	$R_1$	$R_2$	$R_3$	$R_4$	$R_1$	$R_2$	$R_3$	$R_4$
A	1	1	1	1	4	0	0	2
B	0	1	0	0	10	1	4	4
C	3	2	0	1	6	1	1	3
D	1	2	1	0	6	5	0	0
E	2	1	2	1	8	3	0	4

3. Proposer un algorithme pour déterminer si un état est sain.

**Exercice 2.**

*Graphes d'allocation de ressource*

Réduire les graphes d'allocation de ressource suivants pour détecter s'il y a un interblocage.

1.

2.