# BANKER'S ALGORITHM

BY THANAPOL SOPANHARI 6113632

## DATA STRUCTURES

- Let 'n' be the number of processes in the system and 'm' be the number of resources types.
- Available :
- It is a 1-d array of size 'm' indicating the number of available resources of each type.
- Available[j] = k means there are **'k'** instances of resource type **R**<sub>i</sub>
- Max :
- It is a 2-d array of size '**n\*m'** that defines the maximum demand of each process in a system.
- Max[i, j] = k means process P<sub>i</sub> may request at most 'k' instances of resource type R<sub>j</sub>.

### DATA STRUCTURES

- Allocation :
- It is a 2-d array of size 'n\*m' that defines the number of resources of each type currently allocated to each process.
- Allocation[i, j] = k means process P<sub>i</sub> is currently allocated 'k' instances of resource type R<sub>j</sub>
- Need :
- It is a 2-d array of size **'n\*m'** that indicates the remaining resource need of each process.
- Need [i, j] = k means process P<sub>i</sub> currently need 'k' instances of resource type R<sub>j</sub>
- for its execution.
- Need [ i, j ] = Max [ i, j ] Allocation [ i, j ]

### SAFETY ALGORITHM

- 1) Let Work and Finish be vectors of length 'm' and 'n' respectively.
   Initialize: Work = Available
   Finish[i] = false; for i=1, 2, 3, 4....n
- 2) Find an i such that both

   a) Finish[i] = false
   b) Need<sub>i</sub> <= Work</li>
   if no i exists go to step (4)
- 3) Work = Work + Allocation[i] Finish[i] = true goto step (2)
- 4) if Finish [i] = true for all i then the system is in a safe state

#### PROGRAM IMPLEMENTATION

First step is to put your resource and process sizes.

Enter resource size: 3 Enter process size: 5

#### THERE ARE 2 MODES

# Manual mode

# Random mode

Press M for manual inout or any to random input: Put resource input:

R1 = 10

R2 =

R3 =

# MANUAL MODE

PRESS M TO SELECT THE MANUAL MODE

#### Put allocation input:

010

200

302

211

0 0 2

Put max input:

753

3 2 2

902

222

433



R0 = 10 $R1 = 5$ $R2 = 7$ $R1$ $R2$ $R3$ $R1$	CON ALLO 3 R1 F P0 0 1	ION- R3 0	loca r2	AL		R2 = 7	1 = 5	0 R	0 = 1	RC
R1       R2       R3       R1       R2       R3       P0       0       1       0       P0       0       1       0       P1       0       0       P0       0       P1       0       P1       0       P2       3       0       P1       0       P2       3       0       P1       0       0       P2       6       0       P2       6       0       P2       6       0       P2       6       0       P2       1       P2       0       P2       1       P2	R3 R1 F P0 0 1	R3 0	R2	D 1						
R1R2R3 $P0$ $0$ $1$ $0$ $P1$ $0$ $0$ P0 $0$ $1$ $0$ $P2$ $3$ $0$ $2$ $P2$ $3$ $0$ $P2$ $3$ $P2$ $0$ $P2$	PO 0 1	0		R1			TON	T.OC'A	77	
R1       R2       R3       P1       0       0       0       0       P1       0       P2       3       0       2       93       0       2       P3       2       1       1       P4       0       0       2       P3       2       1       1       P4       0       0       2       P3       0       P4       0       0       P4       0       0       P4       0       0       P4       0       1       1       1       1       P4       0			1	0	P		"	LOCA	AL	
P0       0       1       0 $P2$ 3       0       2 $P3$ 2       1       1 $P3$ 0 $P4$ 0	P1 0 (	0	0	0	P		R3	R2	R1	
P1       2       0       0 $P^3$ 2       1       1 $P^3$ 0 $P^4$ 0 $P^2$ 0 $P^3$ 0 $P^4$	P2 3 (	2	1	3	P		0	1	0	PO
P2 $3$ $0$ $2$ P4 $0$ P3 $2$ $1$ $1$ $0$ $2$ $R1$ $R2$ $R3$ $R1$ P4 $0$ $0$ $2$ $R1$ $R2$ $R3$ $R1$ $R1$ $R2$ $R3$ $R1$ $R2$ $R3$ $R1$ $R2$ $R3$ $R1$ $R1$ $R2$ $R3$ $R1$ $R1$ $R1$ $R2$ $R3$ $R1$ $R2$ $R3$ $R1$ $R1$ $R1$ $R1$	P3 0 (	1 2	т 0		P.		0	0	2	P1
P3       2       1       1 $R1 R2 R3$ $R1 R2 R3$ $R1 R2 R3$ $R1 R2 R3$ P4       0       0       2 $P0 7 4 3$ $P0 7 4 3$ $R1$ $R1 R2 R3$ $P1 0 0 0$ 0 $P2 6 0 0$ $P3 0 1 1$ $P2 6$ $P3 0 1$ $P4 0$ $P2 6$ $P3 0 1$ $P3 0 1$ $P3 0 1$ $P4 0$ $P4 0$ $P2 6$ $P3 0 1$ $P3 0 1$ $P3 0 1$ $P3 0 1$ $P4 0$ $P1 0$ $P0 0 0$ $P1 0$ $P1 0$ $P2 3 0 2$ $P4 0$ $P2 3$ $P4 0$ $P4 0$ $P2 3$ $P4 0$	P4 0 0	2.					2	0	3	Р2
P3 $2$ $1$ $1$ $R1$ $R2$ $R3$ $R1$ $R2$ $R3$ $R1$ $P4$ $0$ $0$ $2$ $P0$ $7$ $4$ $3$ $P1$ $0$ $0$ $R1$ $R2$ $R3$ $P1$ $0$ $0$ $P2$ $0$ $0$ $P1$ $0$ $P1$ $3$ $2$ $2$ $P4$ $4$ $3$ $1$ $P3$ $0$ $P3$ $0$ $P3$ $0$ $P3$ $0$ $P4$ $0$ $P2$ $6$ $0$ $P3$ $0$ $P4$ $0$ $P2$ $0$ $P3$ $0$ $P4$ $0$ $P3$ $0$ $P4$ $0$ $P2$ $0$ $P4$ $0$ $P2$ $0$ $P2$ $0$ $P2$ $0$ $P1$ $0$ $P2$ $0$ $P2$ $0$ $P2$ $P3$ $0$ $P1$ $0$ $P2$ $P3$ $0$ $P1$ $0$ $P2$ $P3$ $0$ $P4$ $0$ $P2$ $P3$ <td></td> <td></td> <td>ED</td> <td>NE</td> <td></td> <td></td> <td>-</td> <td>1</td> <td>2</td> <td> D2</td>			ED	NE			-	1	2	 D2
P4       0       0       2 $P0$ 7       4       3 $P1$ 0       0 $P1$ $P0$ 7 $P1$ $P0$ 7 $P1$ $P0$ $P1$ $P0$ $P2$ $6$ $0$ $P1$ $0$ $P2$ $6$ $0$ $P4$ $0$ $P2$ $3$ $P0$ $0$ $P1$ $0$ $0$ $P1$ $0$ $0$ $P2$ $3$ $P3$ $0$ $P1$ $0$ $0$ $P2$ $3$ $P3$ $0$ $P1$ $0$ $P2$ $3$	3NEEI	R3	R2	R1				1	-	F.3
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	R1 I		4	7	P		2	0	0	P4
MAX $P2$ $6$ $0$ $P1$ $0$ R1       R2       R3 $P4$ $3$ $1$ $P2$ $6$ P0       7 $5$ $3$ $P4$ $4$ $3$ $1$ $P2$ $6$ $0$ $P1$ $0$ $P2$ $6$ $P3$ $0$ $1$ $1$ $P2$ $6$ $P3$ $0$ $P4$ $0$ $P1$ $0$ $P0$ $P1$ $0$ $P0$ $P1$ $0$ $P1$ $0$ $P2$ $3$ $0$ $2$ $P3$ $0$ $P1$ $0$ $P2$ $3$ $0$ $P2$ $3$ $0$ $P2$ $3$ $P3$ $0$ $P4$ $0$ $P2$ $3$ $P2$ $3$ $P2$ $7$ $7$ $7$ $7$	P0 7 4	0	0	0	P					
R1       R2       R3 $p3$ $0$ $1$ $1$ $p2$ $6$ P0       7 $5$ $3$ $p4$ $4$ $3$ $1$ $p4$ $3$ $1$ $p4$ $3$ $1$ $p3$ $0$ $1$ $1$ $p4$ $0$ $p3$ $0$ $1$ $1$ $p4$ $0$ $p4$ $0$ $p4$ $0$ $p4$ $1$ $p6$ $0$ $1$ <td></td> <td>0</td> <td>0</td> <td></td> <td>P</td> <td></td> <td></td> <td>x</td> <td>ма</td> <td></td>		0	0		P			x	ма	
N1       N3 $p_4$ $q_4$ $q_3$ $p_3$ $p_3$ $p_4$ P0       7       5       3 $p_1$ $q_4$ $q_3$ $p_1$ $p_3$ $p_4$ $q_4$ $q_3$ $p_1$ $p_4$ $q_4$ $q_3$ $p_1$ $p_4$ $q_4$ $q_3$ $p_1$ $p_4$ $q_4$ $q_3$ $p_1$ $p_4$ $q_4$ $q_3$ $q_1$ $p_4$ $q_4$ $q_3$ $q_1$ $p_4$ $q_4$ $q_3$ $q_1$ $p_4$ $q_4$ $q_3$ $q_1$ $q_4$ $q_1$	P2 6 (	1	1	0	P		ъз	R2	ъ1	
P0       7       5       3 $P4$ 0 $P1$ 3       2       2 $Available: [5, 3, 2]$ $Available: P4$ $Available:$	P3 0 0	1	3	4	P		2	E .	7	
P1       3       2       2       Availab         P2       9       0       2       R1       R2       R3      A         P3       2       2       2       R1       R2       R3      A       R1         P4       4       3       3       P0       0       1       0       P0       0        NEED       R1       R2       R3       P1       0       0       P2       3       0       0       P1       0       P2       3       P3       0       P1       0       P2       3       P3       0       P4       0       P4<	3. 21	з.	e:[5	ailabl	A	_	3	5		PO
P2       9       0       2 $R1$ $R2$ $R3$ $R1$ $R2$ $R3$ $R1$ $R2$ $R3$ $R1$ $R2$ $R3$ $R1$ $R1$ $R2$ $R3$ $R1$ $R1$ $R2$ $R3$ $R1$ $R1$ $R2$ $R3$ $P0$ $0$ $R1$ $P2$ $3$ $0$ $0$ $P1$ $0$ $0$ $P1$ $0$ $P2$ $3$ $P1$ $0$ $P2$ $3$ $P1$ $0$ $P2$ $3$ $P1$ $0$ $P2$ $3$ $P3$ $0$ $P2$ $3$ $P3$ $0$ $P2$ $3$ $P2$ <td>Available</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>2</td> <td>2</td> <td>3</td> <td>P1</td>	Available						2	2	3	P1
P3       2       2       2       R1       R2       R3      A         P4       4       3       3       P0       0       1       0       R1         P4       4       3       3       P1       0       0       0       P1       0       P0       0        NEED       P3       0       0       0       P1       0       P2       3       0       P3       0       0       P1       0       P2       3       0       P3       0       P4       0       P3       0       P4       0       P3       0       P4	ON	ION-	LOCA	AL			2	0	9	P2
P4       4       3       3       P0       0       1       0       P1       0       P0       0       P1       0       P0       0       P1       0       P1       0       P1       0       P2       3       0       P1       0       P2       3       P1       0       P2       3       P1       0       P2       3       P1       0       P2       3       P3       0       P1       0       P2       3       P3       0       P1       0       P2       3       P3       0       P4       0       P2       3       0       P4       0       P2       3       P3       0       P4       0       P2       3       0       P4       0       P2       3       0       P4       0       P2       3       0       P4       0       P2       1       0       P4       0       P1       0       P1       0       P1       0       P1       0       P1       0       P2       6       P3       0       P1       0       P2       6       P3       0       P2       6       P3       0       P2       6       P3       0	(3ALL)	R3	R2	R1			2	2	2	Р3
P1       0       0       0       P0       0         P2       3       0       2       P1       0       P1       0         P3       0       0       0       2       P1       0       P2       3       P1       0       P2       3       P1       0       P2       3       P1       0       P2       3       P2       3       P2       3       P3       0       P2       3       P3       0       P2       3       P3       0       P2       3       P3       0       P4       0       P2       3       0       P4       0       P3       0       0       P1       0       P1       0       P2       6       P1       0       P2       6       P3       0       P1       0       P2       6       P3       0       P2       6       P3       0       P2       6       P3       0       P2       6       P3       0       P2 <td>R1 I</td> <td>0</td> <td>1</td> <td>0</td> <td>P</td> <td></td> <td>3</td> <td>3</td> <td>4</td> <td>P4</td>	R1 I	0	1	0	P		3	3	4	P4
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	PO 0 (	0	0	•	P					
R1       R2       R3       0       0       0       0       1	P1 0 (	2	0	3	P					
R1       R2       R3       P4       0       0       2       P3       0         P0       7       4       3      NEED       P4       0         P1       1       2       2       R1       R2       R3      NEED	P2 3 (	0	0	0	P.			ED	NE	
P0       7       4       3      NEED       P4       0         P1       1       2       2       R1       R2       R3      N         P2       6       0       0       7       4       3      N       R1         P3       0       1       1       P2       6       0       0       R1         P4       4       3       1       P3       0       0       0       P1       0         P4       4       3       1       P3       0       0       P1       0         P4       4       3       1       P2       6       0       0       P1       0         P4       4       3       1       P3       0       0       P2       6         P3       0       0       0       P2       6       P3       0	P3 0 (		Ŭ	Ŭ,	E.		R3	R2	R1	
P1       1       2       2       R1       R2       R3      N         P2       6       0       0       7       4       3      N         P3       0       1       1       P2       6       0       0       R1         P3       0       1       1       P2       6       0       0       P1       0         P4       4       3       1       P3       0       0       0       P1       0         P4       4       3       1       P2       6       P3       0       P2       6	P4 0 0		ED	NE			3	4	7	РO
P2     6     0     0     7     4     3    N       P3     0     1     1     1     P2     6     0     0       P4     4     3     1     1     P3     0     0     1       P4     4     3     1     1     1     1     1	.3	R3	R2	R1			2	2	1	Р1
P2     6     0     0     R1       P3     0     1     1     P2     6     0     P0     0       P4     4     3     1     P3     0     0     P1     0       P4     4     3     1     P2     6     0     0     P1     0       P4     4     3     1     P2     6     P3     0	,NEEI		4	7	P		~	~	~	52
P3     0     1     P2     6     0     P0     0       P4     4     3     1     P3     0     0     P1     0       P4     4     3     1     P2     6     0     0     P1     0       P4     4     3     1     P2     6     P3     0       P4     4     3     1     P2     6       P3     0     1     1     P2     6       P3     0     1     1     1     1	R1 F	0	0	• •	P		0	0		22
P4     4     3     1     P3     0     0     P1     0       P4     4     3     1     P2     6       P3     0     0     0     1     0	PO 0 0	0	0		P		1	1	0	Р3
P4 4 3 1 P2 6 P3 0	P1 0 0	0	0	0	P:		1	3	4	P4
P3 0	P2 6 0	T	3	4	P					
$\Delta vallable   1 3 2   2 vallable   2 4 3  $	P3 0 0	4	a • 17	ailabl		21	2 2	a. 1	ilabl	Awai
Available: [7, 4, 5] P4 0	P4 0 0	- 1	0.17	arrapi	A	21	SI SI	o. (	LIADI	Aval

ATION						
R3		ALI	LOCA	FION		
0		ъ1	ъ2	ъЭ		
0		KT	R2	кэ		
2	PO	0	0	0		
0	P1	0	0	0		
0						
	P2	0	0	0		
R3	P3	0	0	0		
3	ъл	0	0	0		
0	F7	U	U	U		
0						
0		NEI	ED			
Ŭ 🗖						
7, 4, 5]		R1	R2	R3		
	PO	0	0	0		
ATION	<b>n</b> 1	^	•	0		
R3	FI	U	U	U		
0	P2	0	0	0		
0	Dζ	٥	٥	0		
0	<b>F</b> 0					
0	P4	0	0	0		
			F 41 /			
R3	Avaı.	Lable	9:[T(	J <b>,</b> 5,	/]	
0						
0	7	1-61-		ator.	ъ	<u>.</u>
0	Aval.	rapie	e ve	ctor:	R	0.

R2

The safe-state order: <1>, <3>, <4>, <0>, <2>

Press M for manual inout or any to random input: Rando

R0 = 11 R1 = 14 R2 = 17

R3 15

11 17

11

-----ALLOCATION------

	-	
P1	2	
P2	1	
P3	4	
P4	3	2
	ма	x-
	R1	I
PO	4	4
P1	7	4
P2	2	
ΡЗ	6	(
P4	4	

NEED							
	R1	R2	R3				
РO	3	7	12				
P1	5	2	12				
Р2	1	12	10				
Р3	2	5	16				
Р4	1	11	10				

Available: [0, 1, 7] The system is unsafe or deadlock occurs! <-1> -> <-1> -> <-1> -> <-1> -> <-1>

# RAMDOM MODE

PRESS ANY KEYS TO SELECT THE MANUAL MODE

#### SAFE ALGORITHM

```
fun isSafe(): Boolean {
   var isVisited = Array(processSize) { false }
   var work = createAvailable()
```

```
if (count == processSize)
    print("\nAvailable vector: ")
    for (i in 0 until resourceNum.size) {
       print(" R$i = ${resourceNum[i]} ")
   print("\nThe safe-state order: ")
    var result = "<${resultSequence[0]}>"
    for (i in 1 until resultSequence.size) {
        result += ", <${resultSequence[i]}>"
   print(result)
} else {
   println("The system is unsafe or deadlock occurs!")
   var result = "<${resultSequence[0]}>"
    for (i in 1 until resultSequence.size) {
        result += " -> <${resultSequence[i]}>"
   println(result)
    return false
```

#### RANDOM INOUT

```
fun generateMax() { // Each value less than or equal Ri
```

```
for (column in 0 until resourceSize) {
    for (row in 0 until processSize) {
        try {
            max[row][column] = random.nextInt(resourceNum[column]) + 1
        } catch (e: Exception) {
            print(e)
            }
        }
```

fun generateAllocation() { // Sum of each resource in all process
var maxInColumn = 0

for (column in 0 until resourceSize) {
 for (row in 0 until processSize) {
 maxInColumn += max[row][column]

```
hile (true) {
var <u>arrayEachResource</u> = Array(<u>processSize</u>) {
<u>random</u>.nextInt(<u>max[it][column]) + 1</u>
```

```
if (arrayEachResource.sum() <= resourceNum[column]) {
    putArrayInEachResource(arrayEachResource, column)
    break</pre>
```

### RANDOM INPUT



# THANK YOU