

# Project GPU : Path merge

Arthur **Zucker** , Clément **Apavou**



13 décembre 2020

# Table des matières

## ① Introduction



## Keywords

## CUDA · Stream · Merging

- We have to finish the code
- Then work on the report
- Finally work on the beamer

# Path merged : thread 0

		B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
		4	7	8	10	12	13	14
A[0]	1	1	1	1	1	1	1	1
A[1]	2	1	1	1	1	1	1	1
A[2]	5	0	1	1	1	1	1	1
A[3]	6	0	1	1	1	1	1	1
A[4]	6	0	1	1	1	1	1	1
A[5]	9	0	0	0	1	1	1	1
A[6]	11	0	0	0	0	1	1	1
A[7]	15	0	0	0	0	0	0	0
A[8]	16	0	0	0	0	0	0	0

Diagram illustrating a merged path for thread 0. A red vertical line labeled 'y' and a red horizontal line labeled 'x' intersect at the cell (A[0], B[0]). Dashed diagonal lines represent the path of the thread.

1 Calcul de P et K :

# Path merged : thread 0

		B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
		4	7	8	10	12	13	14
A[0]	1	1	1	1	1	1	1	1
A[1]	2	1	1	1	1	1	1	1
A[2]	5	0	1	1	1	1	1	1
A[3]	6	0	1	1	1	1	1	1
A[4]	6	0	1	1	1	1	1	1
A[5]	9	0	0	0	1	1	1	1
A[6]	11	0	0	0	0	1	1	1
A[7]	15	0	0	0	0	0	0	0
A[8]	16	0	0	0	0	0	0	0

Diagram illustrating a path merged for thread 0. A red vertical line is drawn between the first column (A indices) and the second column (B indices). A red horizontal line is drawn between the second row (B indices) and the third row (A indices). A red 'X' is placed at the intersection of these lines. A red 'y' is placed at the bottom of the vertical line, and a red 'x' is placed at the right end of the horizontal line. Dashed diagonal lines are drawn across the grid.

1 Calcul de P et K :  
→  $P=(0,0)$

# Path merged : thread 0

		B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
		4	7	8	10	12	13	14
A[0]	1	1	1	1	1	1	1	1
A[1]	2	1	1	1	1	1	1	1
A[2]	5	0	1	1	1	1	1	1
A[3]	6	0	1	1	1	1	1	1
A[4]	6	0	1	1	1	1	1	1
A[5]	9	0	0	0	1	1	1	1
A[6]	11	0	0	0	0	1	1	1
A[7]	15	0	0	0	0	0	0	0
A[8]	16	0	0	0	0	0	0	0

Diagram illustrating a path merged for thread 0. A red vertical line labeled 'y' and a red horizontal line labeled 'x' intersect at the cell (A[0], B[0]). Dashed diagonal lines represent the path of the thread.

1 Calcul de P et K :  
 →  $P=(0,0)$   
 →  $K=(0,0)$

# Path merged : thread 0

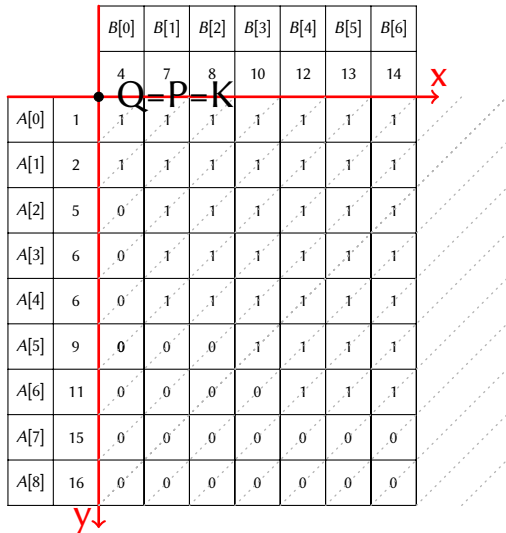
		B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
		4	7	8	10	12	13	14
A[0]	1	1	1	1	1	1	1	1
A[1]	2	1	1	1	1	1	1	1
A[2]	5	0	1	1	1	1	1	1
A[3]	6	0	1	1	1	1	1	1
A[4]	6	0	1	1	1	1	1	1
A[5]	9	0	0	0	1	1	1	1
A[6]	11	0	0	0	0	1	1	1
A[7]	15	0	0	0	0	0	0	0
A[8]	16	0	0	0	0	0	0	0

Diagram illustrating the merged path for thread 0. A red vertical line marks the start of the path at column B[0]. A red horizontal line marks the start of the path at row A[0]. A red 'X' marks the intersection at (A[0], B[0]). Dashed diagonal lines represent the path of the thread.

- 1 Calcul de P et K :  
→  $P=(0,0)$   
→  $K=(0,0)$
- 2 offset = 0



# Path merged : thread 0



1 Calcul de P et K :

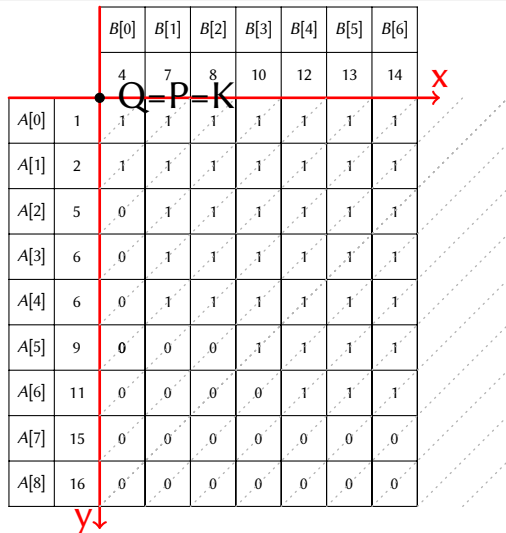
→  $P=(0,0)$

→  $K=(0,0)$

2 offset = 0

3  $Q = (0,0)$

# Path merged : thread 0



- 1 Calcul de P et K :  
→  $P=(0,0)$   
→  $K=(0,0)$
- 2 offset = 0
- 3  $Q = (0,0)$
- 4  $A[Q_y] < B[Q_x]$

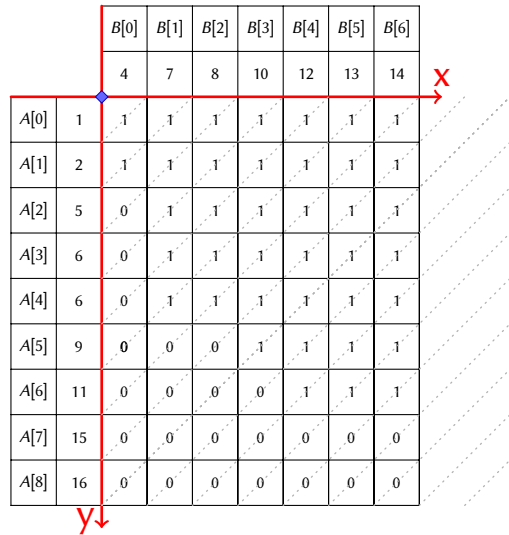
# Path merged : thread 0

		B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
		4	7	8	10	12	13	14
A[0]	1	1	1	1	1	1	1	1
A[1]	2	1	1	1	1	1	1	1
A[2]	5	0	1	1	1	1	1	1
A[3]	6	0	1	1	1	1	1	1
A[4]	6	0	1	1	1	1	1	1
A[5]	9	0	0	0	1	1	1	1
A[6]	11	0	0	0	0	1	1	1
A[7]	15	0	0	0	0	0	0	0
A[8]	16	0	0	0	0	0	0	0

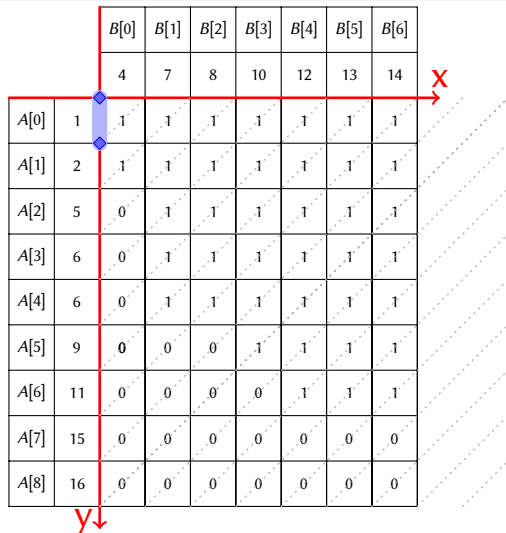
Diagram illustrating the path merged for thread 0. A red dot marks the starting point at A[0][0]. A red arrow labeled 'X' points right from the dot, and a red arrow labeled 'Y' points down from the dot. The text 'Q=P=K' is written near the dot. Dashed diagonal lines represent the path of the thread.

- 1 Calcul de P et K :  
→  $P=(0,0)$   
→  $K=(0,0)$
- 2 offset = 0
- 3  $Q = (0,0)$
- 4  $A[Q_y] < B[Q_x]$
- 5  $M[i] = A[Q_y]$

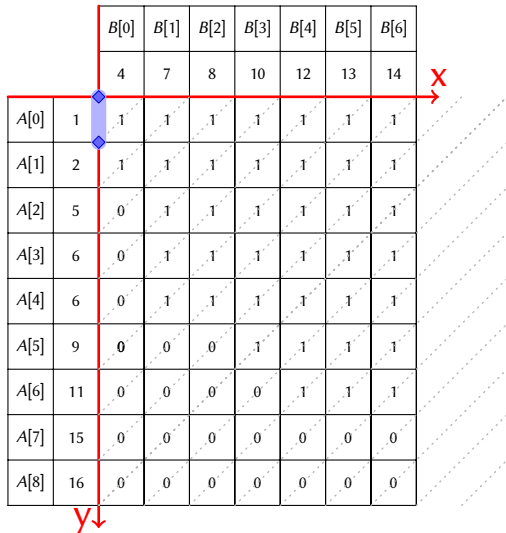
# Path merged : thread 0,path



# Path merged : thread 0,path

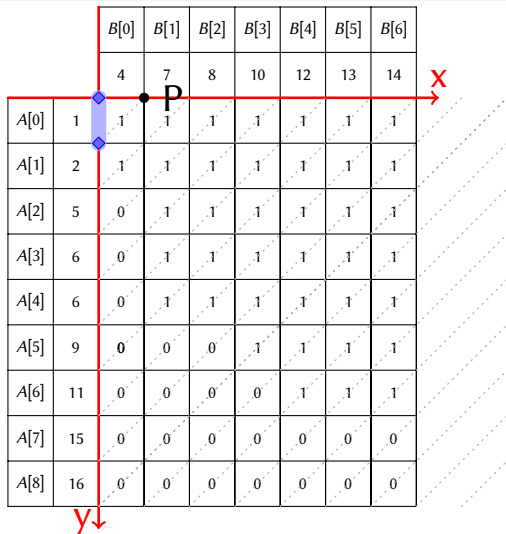


# Path merged : thread 1



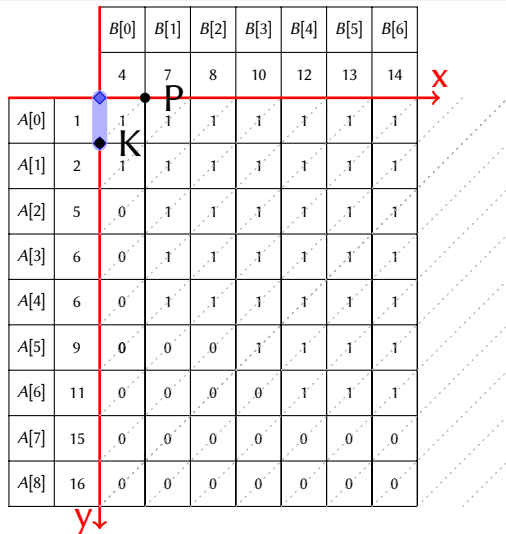
1 Calcul de P et K :

# Path merged : thread 1



1 Calcul de P et K :  
→  $P=(1,0)$

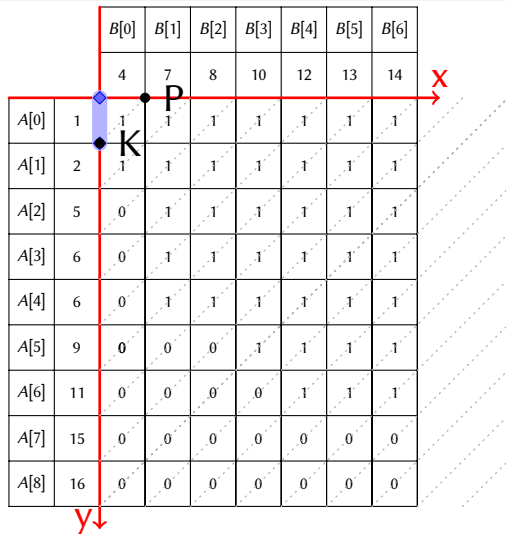
# Path merged : thread 1



- 1 Calcul de P et K :  
→  $P=(1,0)$   
→  $K=(0,1)$

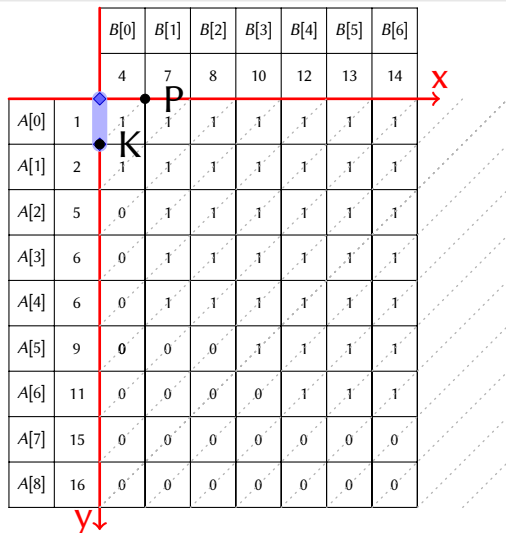


## Path merged : thread 1



- 1 Calcul de P et K :  
→  $P=(1,0)$   
→  $K=(0,1)$
- 2 offset = 0

# Path merged : thread 1



1 Calcul de P et K :

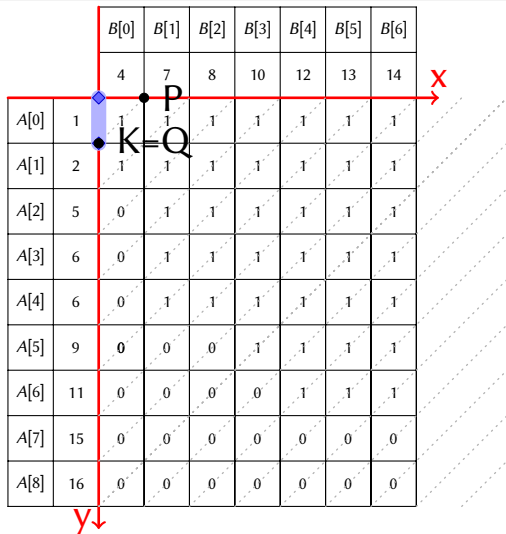
→  $P=(1,0)$

→  $K=(0,1)$

2 offset = 0

3  $Q = (0,1)$

# Path merged : thread 1



1 Calcul de P et K :

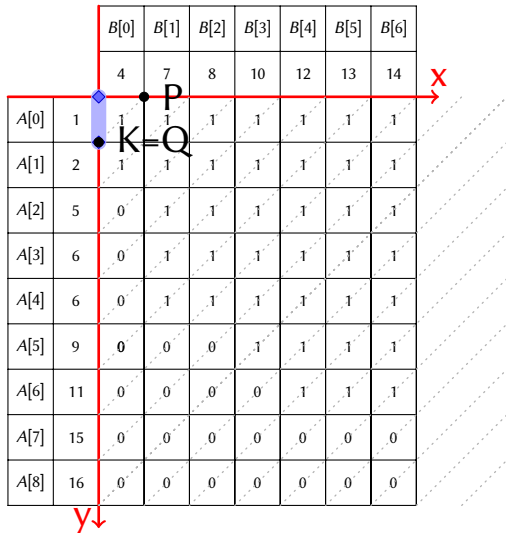
→  $P=(1,0)$

→  $K=(0,1)$

2 offset = 0

3  $Q = (0,1)$

# Path merged : thread 1



1 Calcul de P et K :

→  $P=(1,0)$

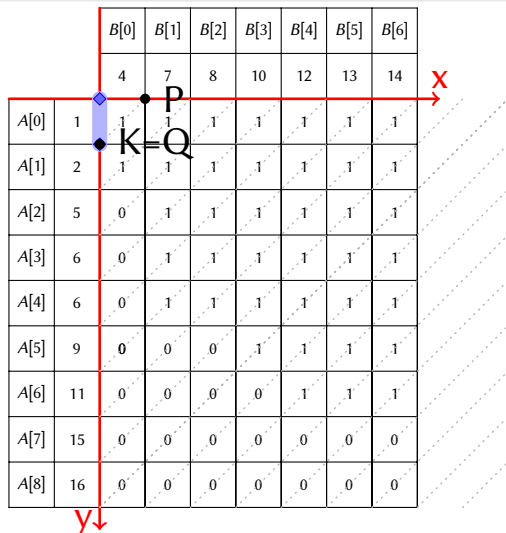
→  $K=(0,1)$

2 offset = 0

3  $Q = (0,1)$

4  $A[Q_y] < B[Q_x]$

# Path merged : thread 1



1 Calcul de P et K :

→  $P=(1,0)$

→  $K=(0,1)$

2 offset = 0

3  $Q = (0,1)$

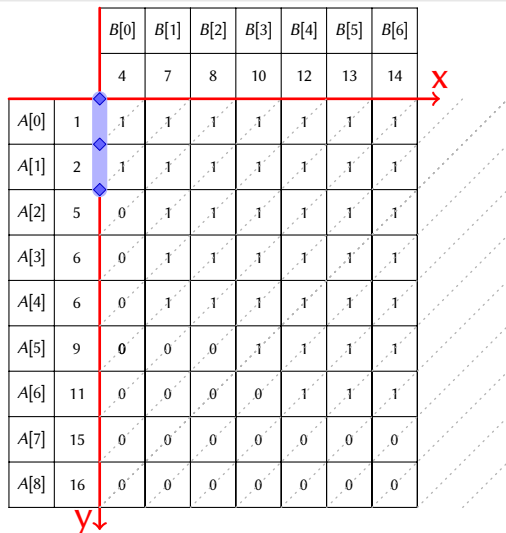
4  $A[Q_y] < B[Q_x]$

5  $M[i] = A[Q_y]$

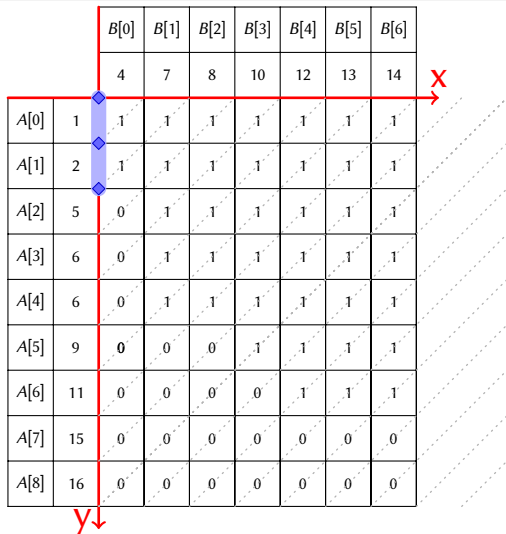
# Path merged : thread 1, path



# Path merged : thread 1, path



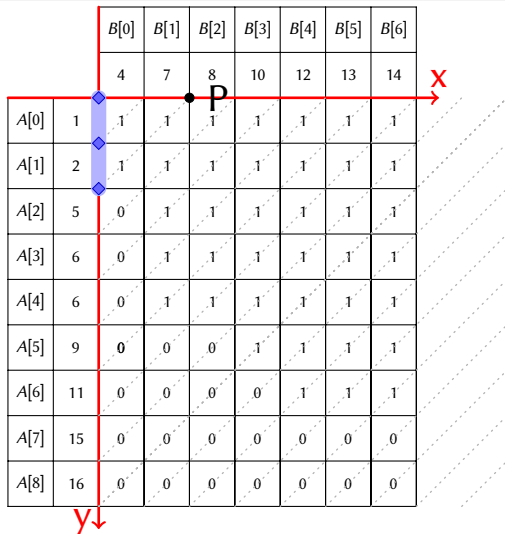
# Path merged : thread 2



1 Calcul de P et K :

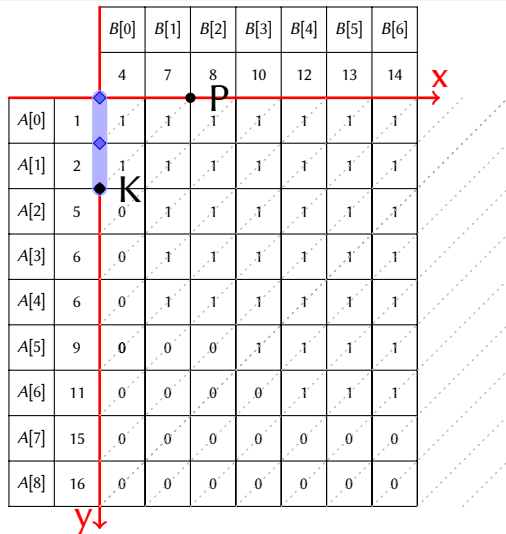


## Path merged : thread 2



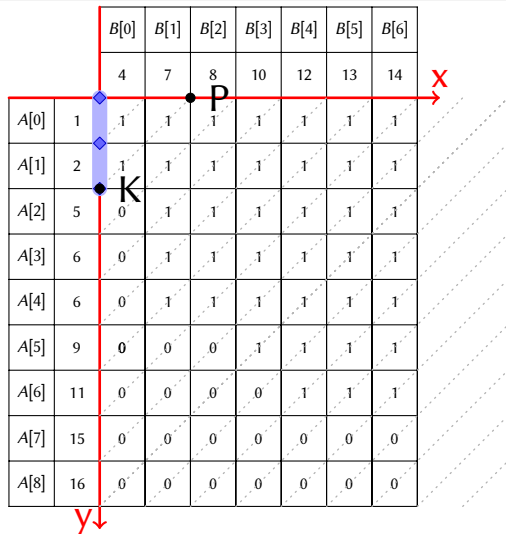
1 Calcul de P et K :  
→  $P=(2,0)$

## Path merged : thread 2



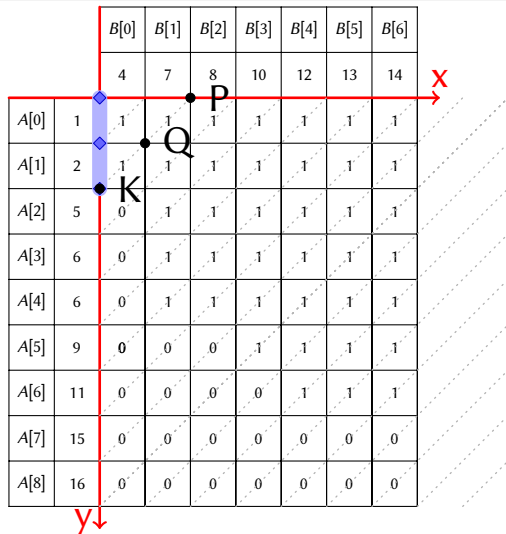
- Calcul de P et K :  
 $\rightarrow P=(2,0)$   
 $\rightarrow K=(0,2)$

## Path merged : thread 2



- 1 Calcul de P et K :  
→  $P=(2,0)$   
→  $K=(0,2)$
- 2 offset = 1

## Path merged : thread 2



1 Calcul de P et K :

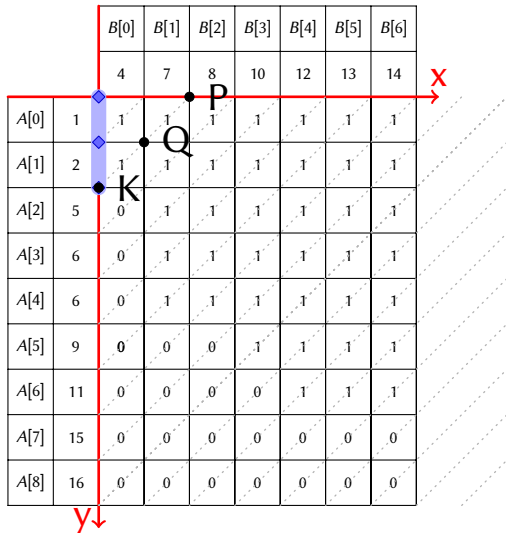
→  $P=(2,0)$

→  $K=(0,2)$

2 offset = 1

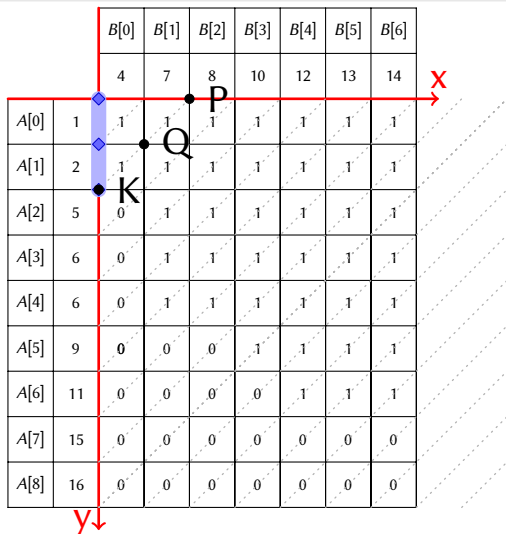
3  $Q = (1,1)$

## Path merged : thread 2



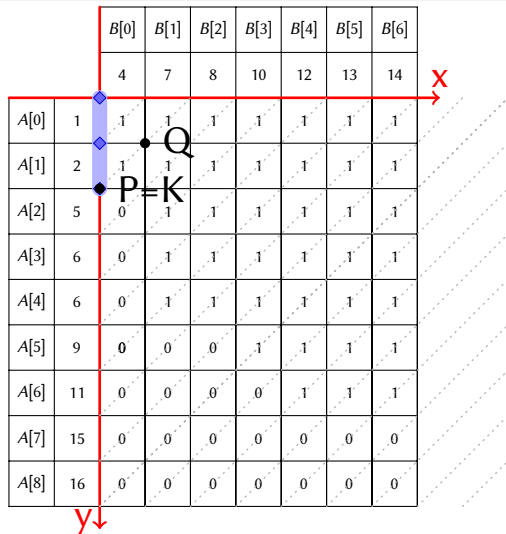
- 1 Calcul de P et K :  
 $\rightarrow P=(2,0)$   
 $\rightarrow K=(0,2)$
- 2 offset = 1
- 3  $Q = (1,1)$
- 4  $P = (Q_x-1, Q_y+1)=(0,2)$

## Path merged : thread 2



- 1 Calcul de P et K :  
 $\rightarrow P=(2,0)$   
 $\rightarrow K=(0,2)$
- 2 offset = 1
- 3  $Q = (1,1)$
- 4  $P = (Q_x-1, Q_y+1)=(0,2)$

## Path merged : thread 2



1 Calcul de P et K :

→  $P=(2,0)$

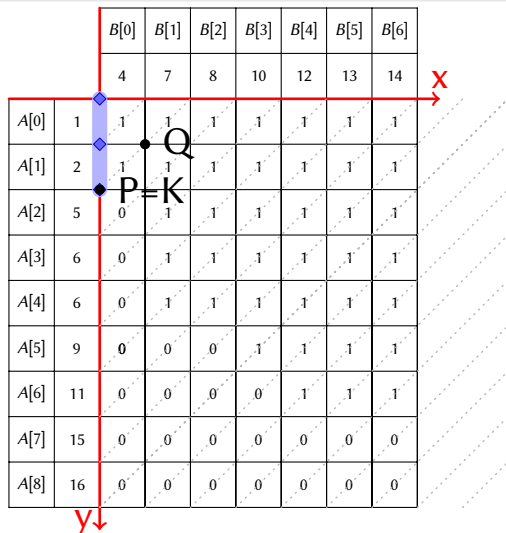
→  $K=(0,2)$

2 offset = 1

3  $Q = (1,1)$

4  $P = (Q_x-1, Q_y+1)=(0,2)$

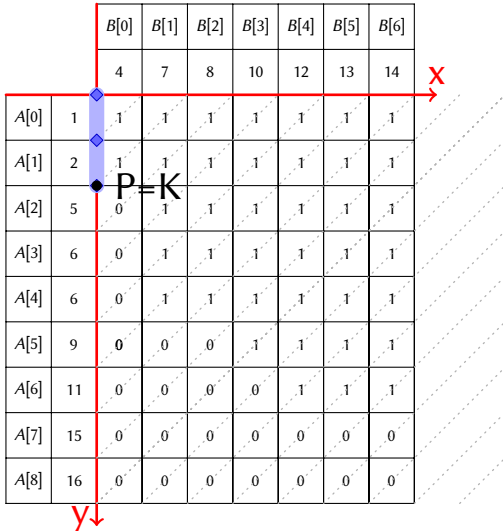
# Path merged : thread 2



- 1 Calcul de P et K :  
→  $P=(2,0)$   
→  $K=(0,2)$
- 2 offset = 1
- 3  $Q = (1,1)$
- 4  $P = (Q_x-1, Q_y+1)=(0,2)$
- 5 On recommence

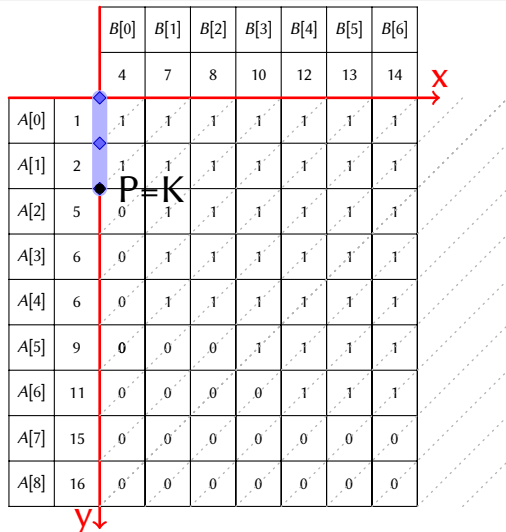


## Path merged : thread 2



$P=K=(0,2)$   
6 offset = 0

# Path merged : thread 2

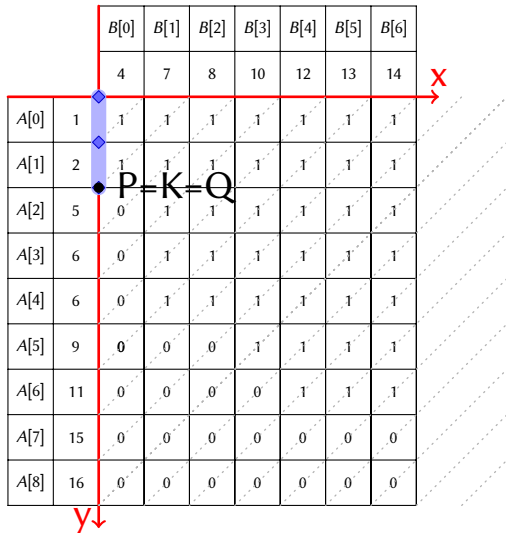


$P=K=(0,2)$

6 offset = 0

7  $Q = (0,2)$

## Path merged : thread 2



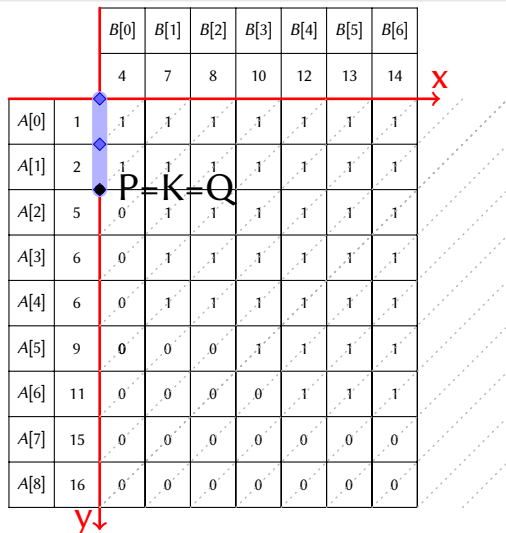
$$P=K=(0,2)$$

6 offset = 0

7  $Q = (0,2)$

8  $A[Q_y] > B[Q_x]$

# Path merged : thread 2



$$P=K=(0,2)$$

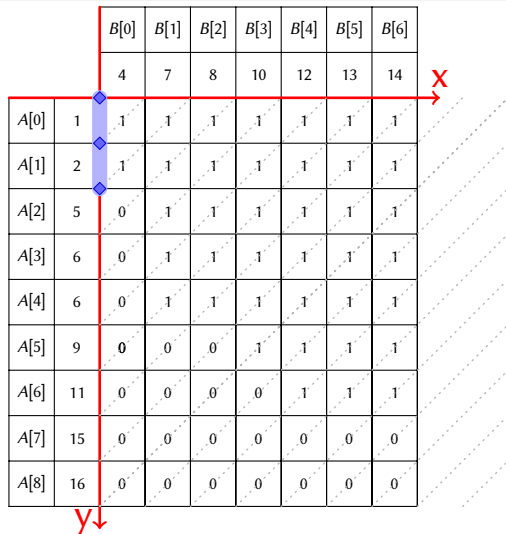
6 offset = 0

7  $Q = (0,2)$

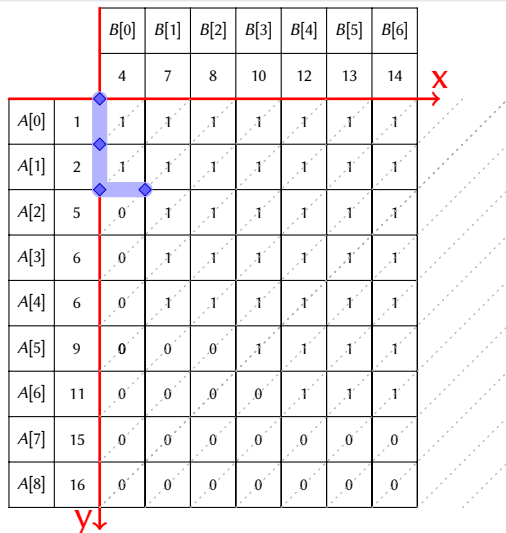
8  $A[Q_y] > B[Q_x]$

9  $M[i] = B[Q_x]$

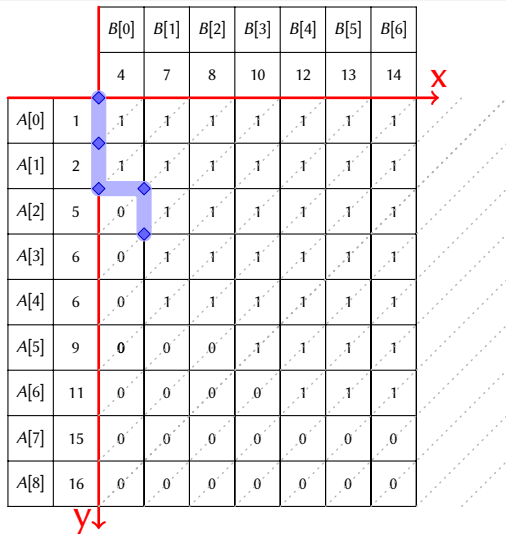
# Path merged : thread 2, path



# Path merged : thread 2, path



# Skipping to thread 9



# Skipping to thread 9

		B[0]	B[1]	B[2]	B[3]	B[4]	B[5]	B[6]
		4	7	8	10	12	13	14
A[0]	1	1	1	1	1	1	1	1
A[1]	2	1	1	1	1	1	1	1
A[2]	5	0	1	1	1	1	1	1
A[3]	6	0	1	1	1	1	1	1
A[4]	6	0	1	1	1	1	1	1
A[5]	9	0	0	0	1	1	1	1
A[6]	11	0	0	0	0	1	1	1
A[7]	15	0	0	0	0	0	0	0
A[8]	16	0	0	0	0	0	0	0

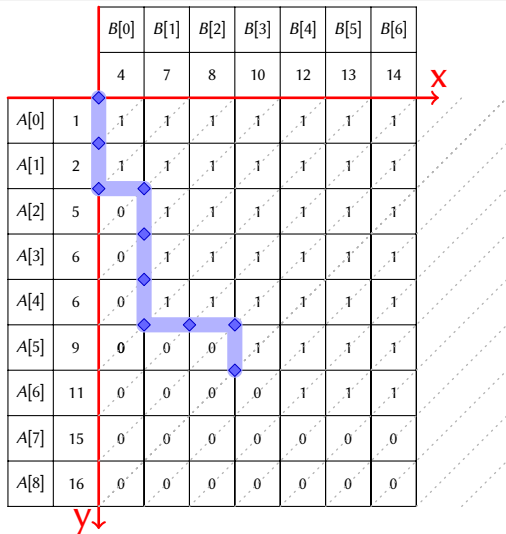
Diagram illustrating a memory access pattern for thread 9. The grid shows values for A[0] through A[8] and B[0] through B[6]. A red 'X' marks the start of the thread's path at A[0][B[0]]. A red arrow points right from A[0][B[0]] to A[0][B[6]]. A red arrow points down from A[0][B[0]] to A[8][B[0]]. A blue path highlights the sequence of memory accesses: A[0][B[0]], A[1][B[0]], A[2][B[0]], A[2][B[1]], A[3][B[1]], A[4][B[1]]. Dashed diagonal lines represent the memory access pattern for other threads.



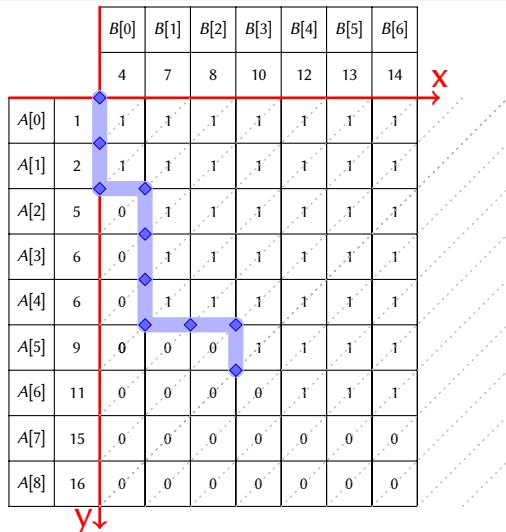
# Skipping to thread 9



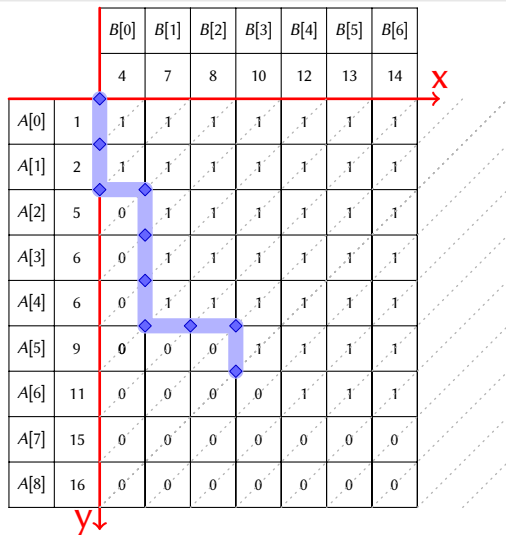
# Skipping to thread 9



# Skipping to thread 9

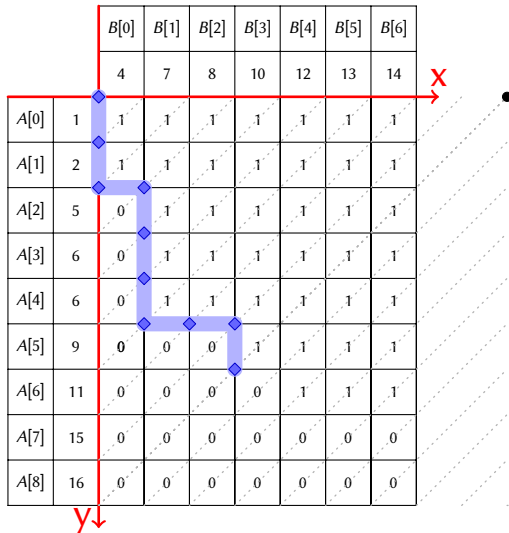


# Path merged : thread 9



1 Calcul de P et K :

## Path merged : thread 9



• P

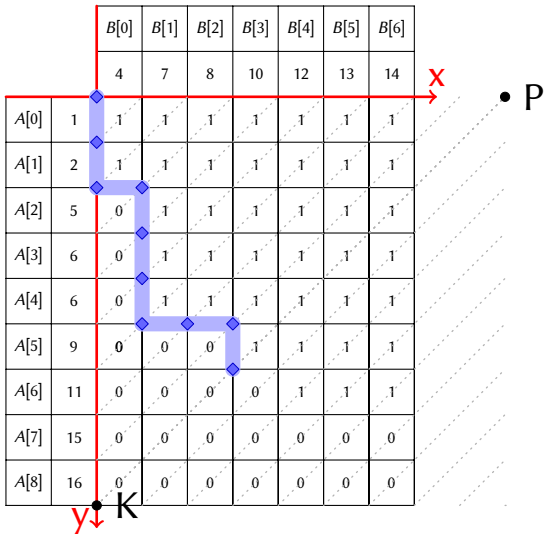
1 Calcul de P et K :  
→  $P=(9,0)$

Diagram illustrating a 2D array  $A$  with dimensions  $9 \times 8$ . The array is divided into four quadrants by a vertical red line at column 2 and a horizontal red line at row 4. The top-left quadrant (columns 0-1, rows 0-3) is shaded light blue. The top-right quadrant (columns 3-7, rows 0-3) is shaded light green. The bottom-left quadrant (columns 0-1, rows 4-8) is shaded light orange. The bottom-right quadrant (columns 2-7, rows 4-8) is shaded light purple. A blue path of diamond markers starts at (0,0), moves right to (1,0), then down to (1,4), then right to (2,4), then down to (2,8), and finally right to (3,8). A red 'X' is at (3,0) and a red 'Y' is at (0,8). A black dot is at (0,0) labeled 'K'.

1 Calcul de P et K :

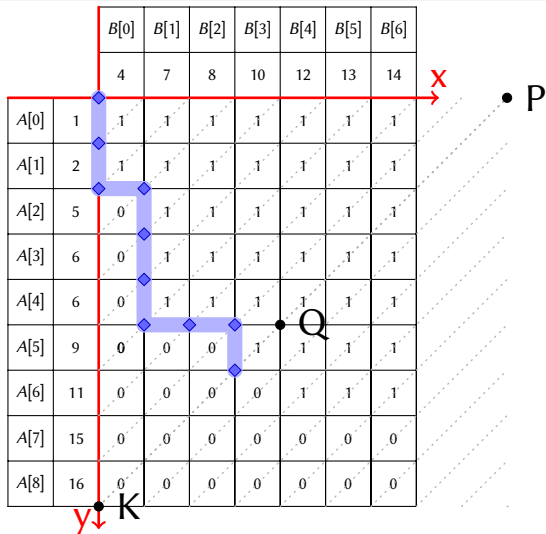
- $P=(9,0)$
- $K=(0,9)$

## Path merged : thread 9



- 1 Calcul de P et K :
  - $P=(9,0)$
  - $K=(0,9)$
- 2 offset = 3

# Path merged : thread 9



1 Calcul de P et K :

→  $P=(9,0)$

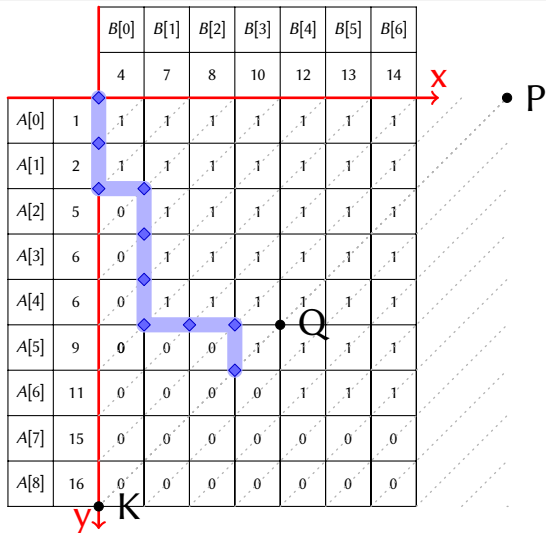
→  $K=(0,9)$

2 offset = 3

3  $Q = (5,4)$



## Path merged : thread 9



1 Calcul de P et K :

→  $P=(9,0)$

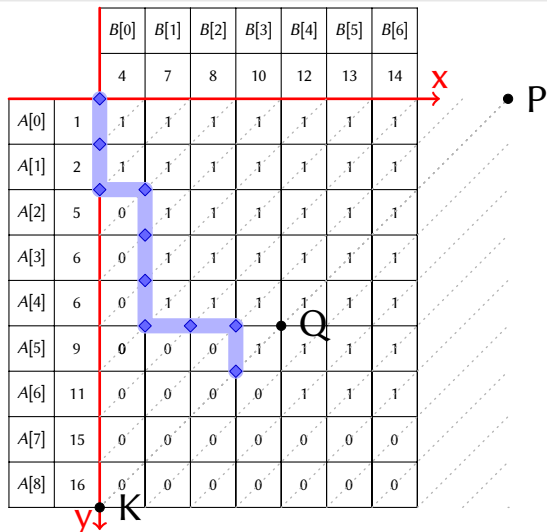
→  $K=(0,9)$

2 offset = 3

3  $Q = (5,4)$

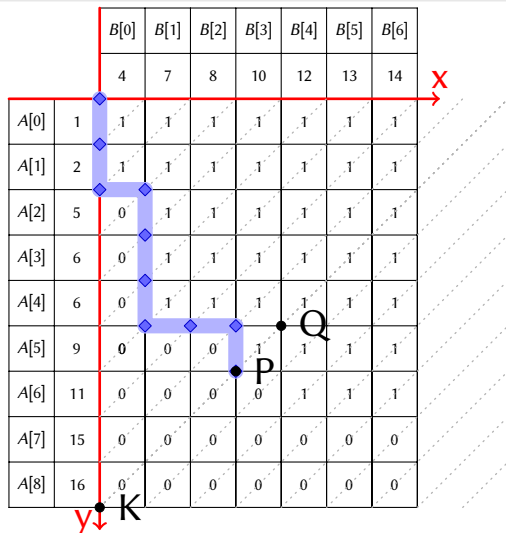
4  $P = (Q_x-1, Q_y+1)=(4,5)$

# Path merged : thread 9



- 1 Calcul de P et K :  
→  $P=(9,0)$   
→  $K=(0,9)$
- 2 offset = 3
- 3  $Q = (5,4)$
- 4  $P = (Q_x-1, Q_y+1)=(4,5)$

# Path merged : thread 9



1 Calcul de P et K :

→  $P=(9,0)$

→  $K=(0,9)$

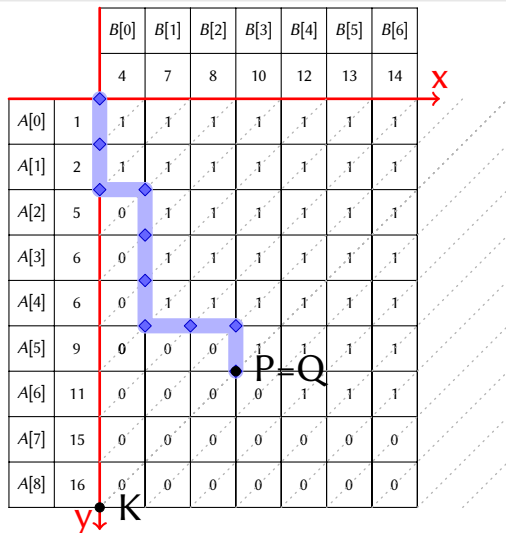
2 offset = 3

3  $Q = (5,4)$

4  $P = (Q_x-1, Q_y+1)=(4,5)$

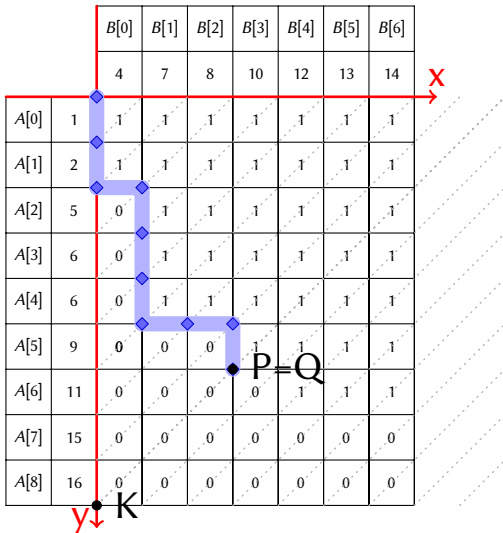
5  $Q = P =(4,5)$

# Path merged : thread 9



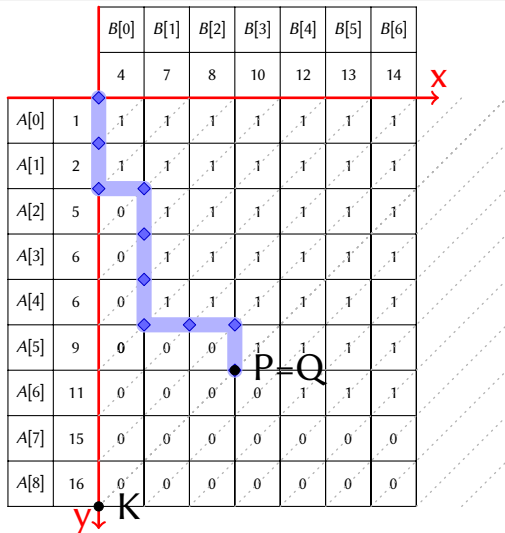
- 1 Calcul de P et K :  
 $\rightarrow P=(9,0)$   
 $\rightarrow K=(0,9)$
- 2 offset = 3
- 3  $Q = (5,4)$
- 4  $P = (Q_x-1, Q_y+1)=(4,5)$
- 5  $Q = P =(4,5)$

## Path merged : thread 9



7  $A[Q_y > B[Q_x]]$

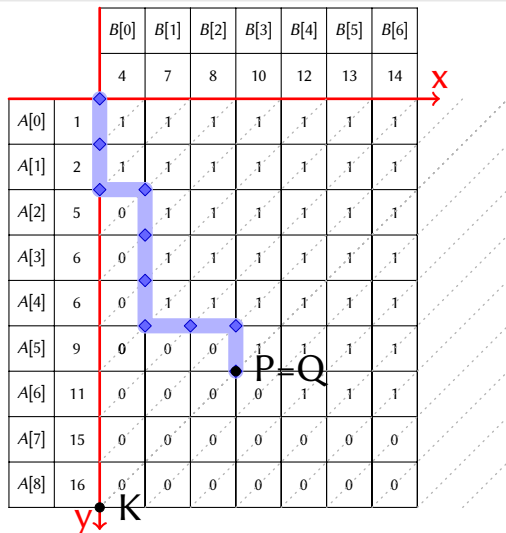
# Path merged : thread 9



7  $A[Q_y > B[Q_x]$

8  $M[9] = B[Q_x]$

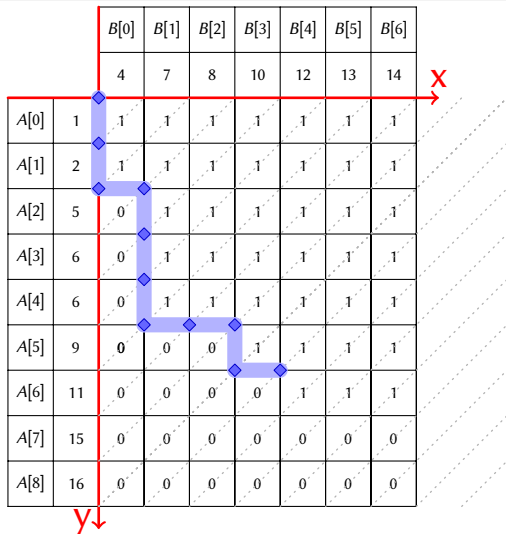
# Path merged : thread 9



7  $A[Q_y > B[Q_x]$

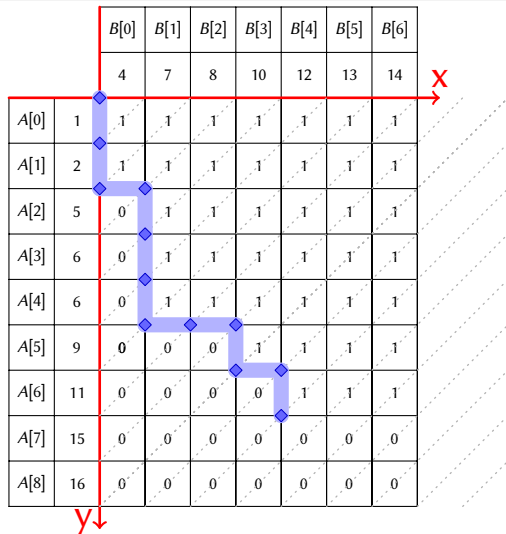
8  $M[9] = B[Q_x]$

# Path merged : thread 9, path

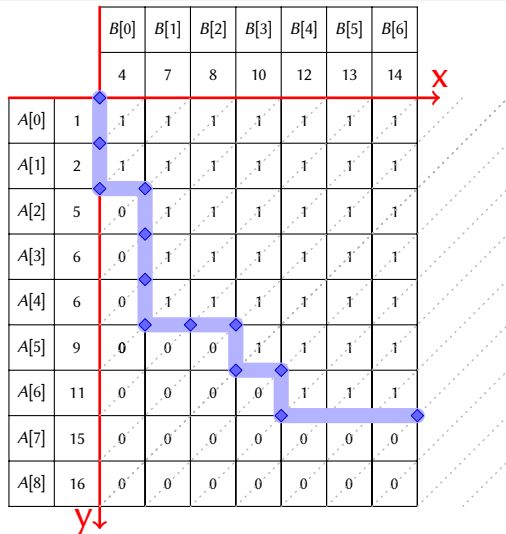




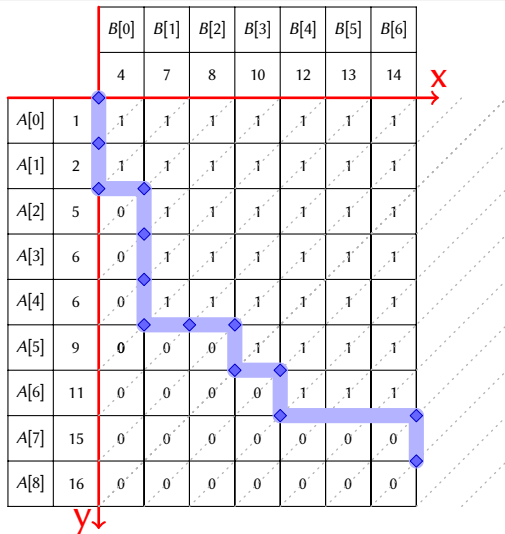
# Path merged : thread 9, path



# Path merged : thread 9, path



Path merged : thread 9, path



# Path merged : thread 9, path

