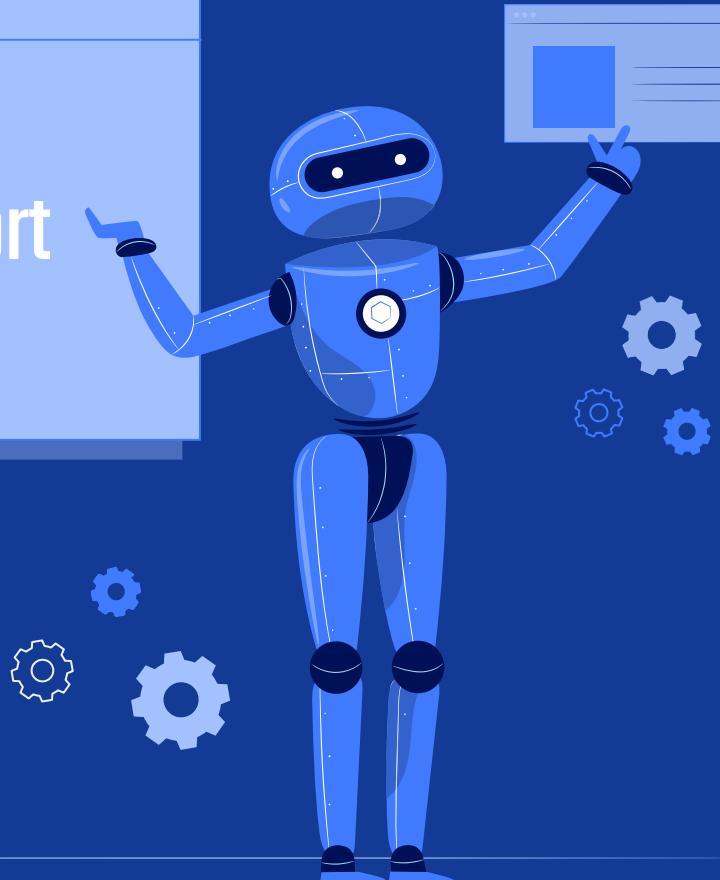


Projet HPC:

Batch merge and merge path sort



Astrid Legay - Marco
Naguib - MAIN5



Sommaire



01

Rappel du sujet



Tri de tableaux



APPLICATIONS

Base de données
Traitement d'image
Théorie des graphes



ALGORITHMES

Tri à bulle
Quicksort
Mergesort

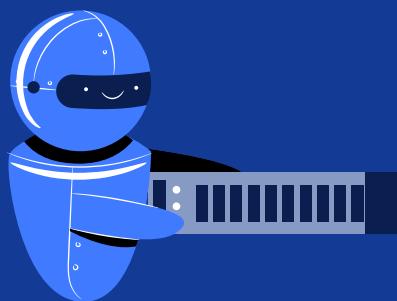


MERGESORT

Très parallélisable
Grâce à la méthode
diviser pour régner

02

Etape 1: mergeSmall_k



ETAPE 1



Pour $|A| + |B| \leq 1024$,
écrire un kernel mergeSmall_k
qui fusionne A et B avec un
block et plusieurs threads.

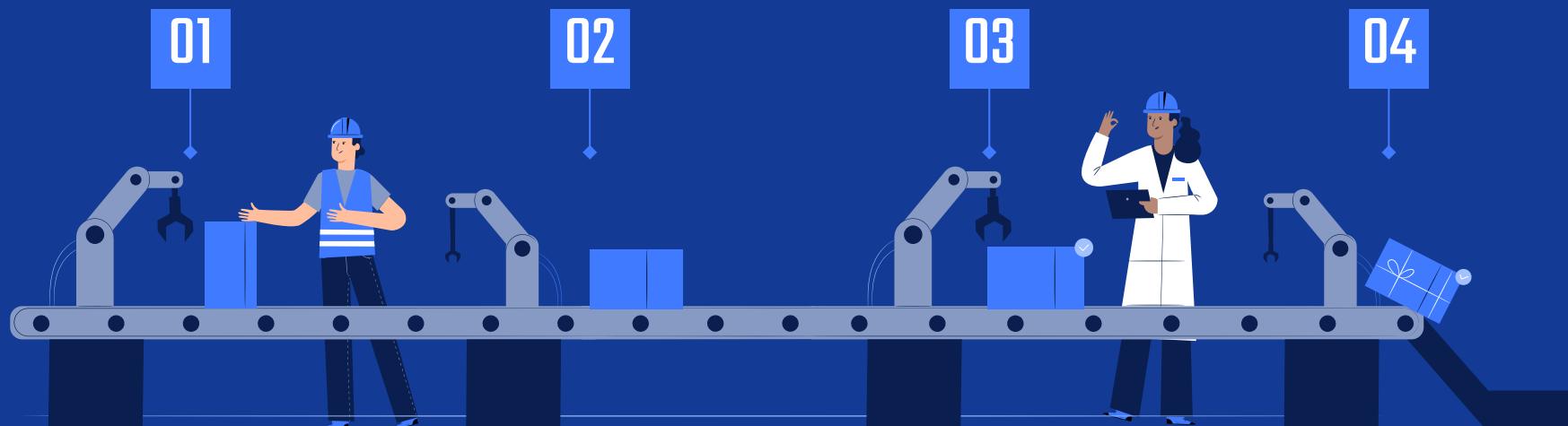
Logique

Ecriture
séquentiel A et
B

Ecriture des
fonctions
utiles

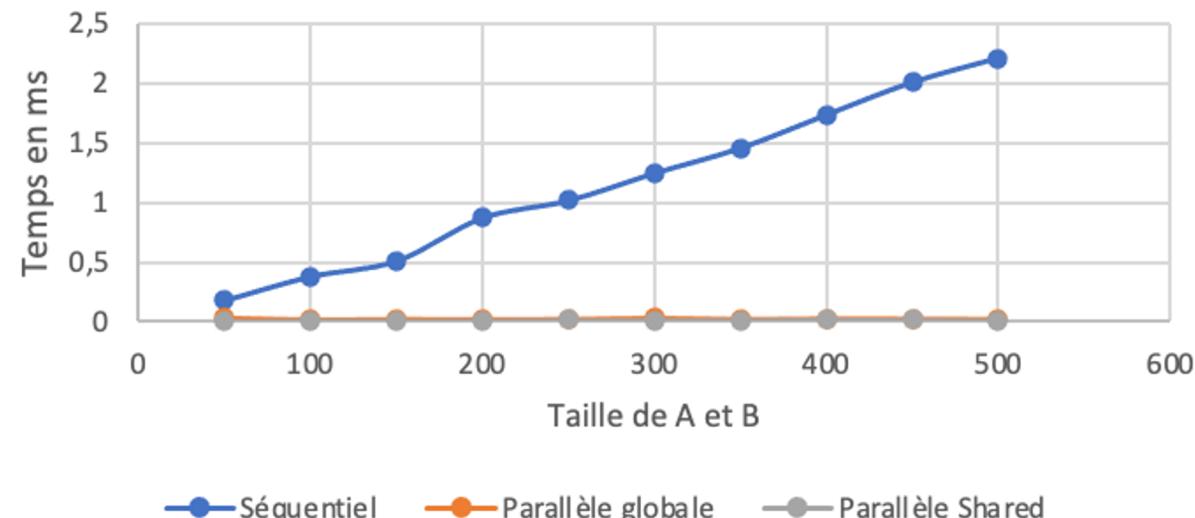
Ecriture en Cuda de
MergeSmall_k sur
mémoire globale et
shared

Tests et mesures

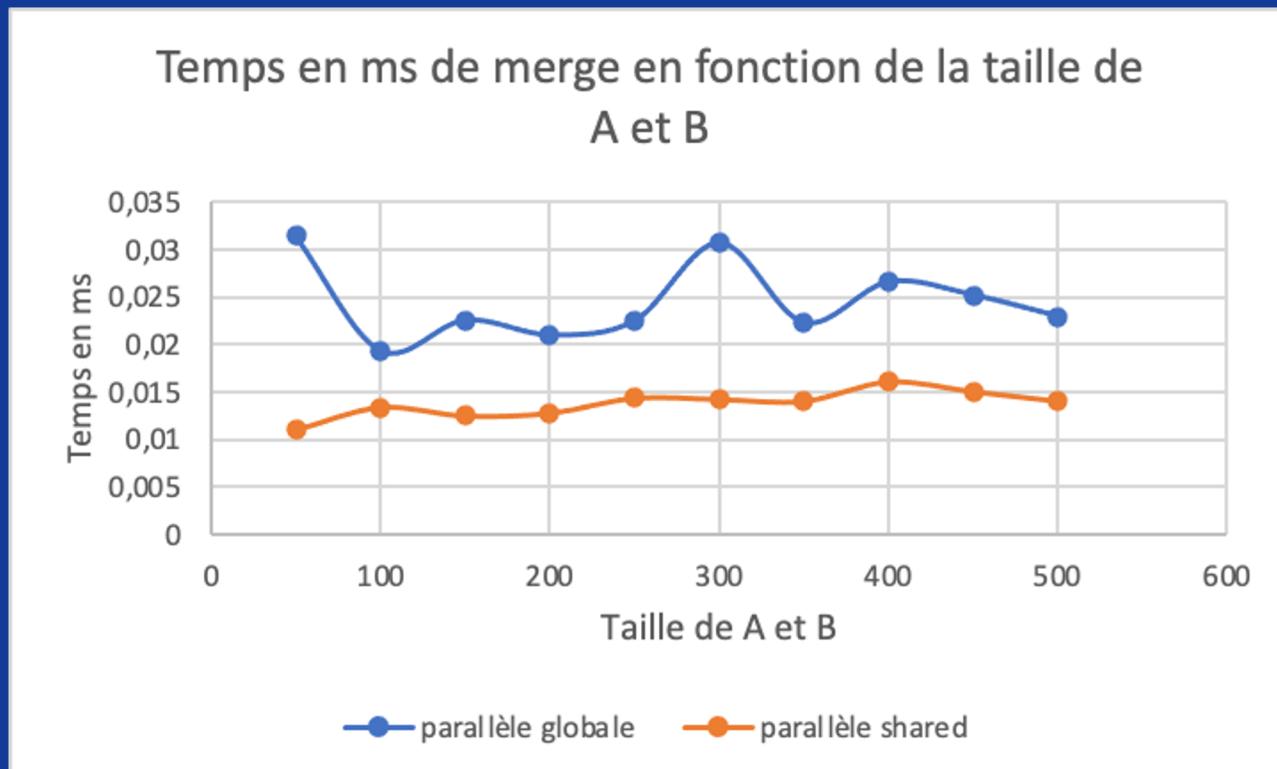


Résultats

Temps en ms de merge en fonction de la taille de A et B



Comparaison de mémoire globale et shared



03

Etape 2 : pathBig_k et mergeBig_k

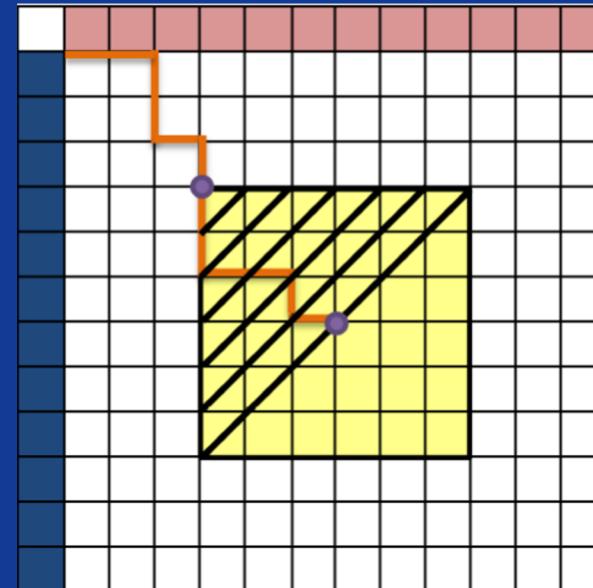
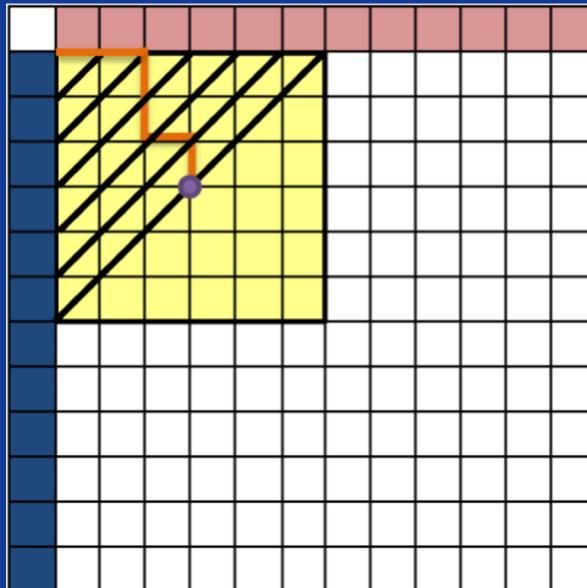


ETAPE 2

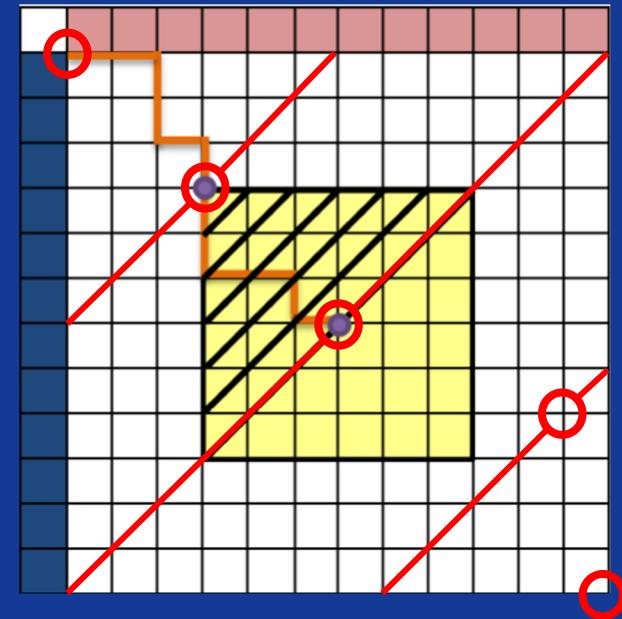
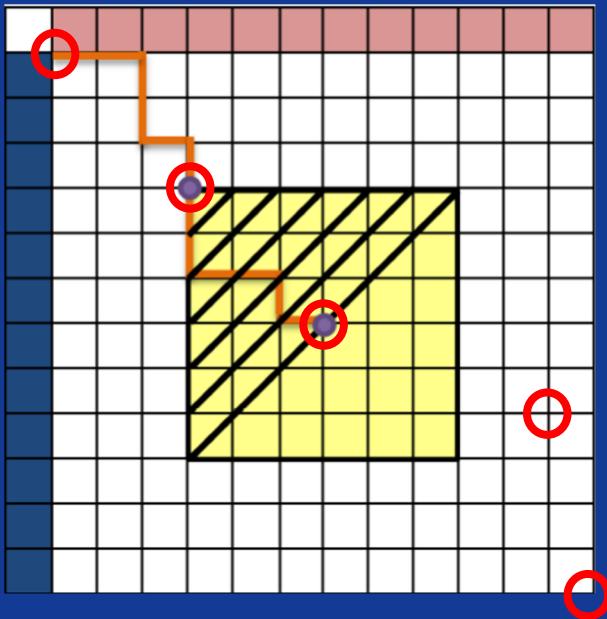


Pour toutes tailles $|A|+|B|=d$, plus petite que la taille de mémoire globale, écrivez deux kernel qui fusionnent et trient A et B en utilisant plusieurs blocks: le 1er kernel pathBig_k qui trouve le chemin et le 2nd mergeBig_k qui fusionne A et B. .

Utilisation des fenêtres glissantes



Utilisation des fenêtres glissantes



Logique du code

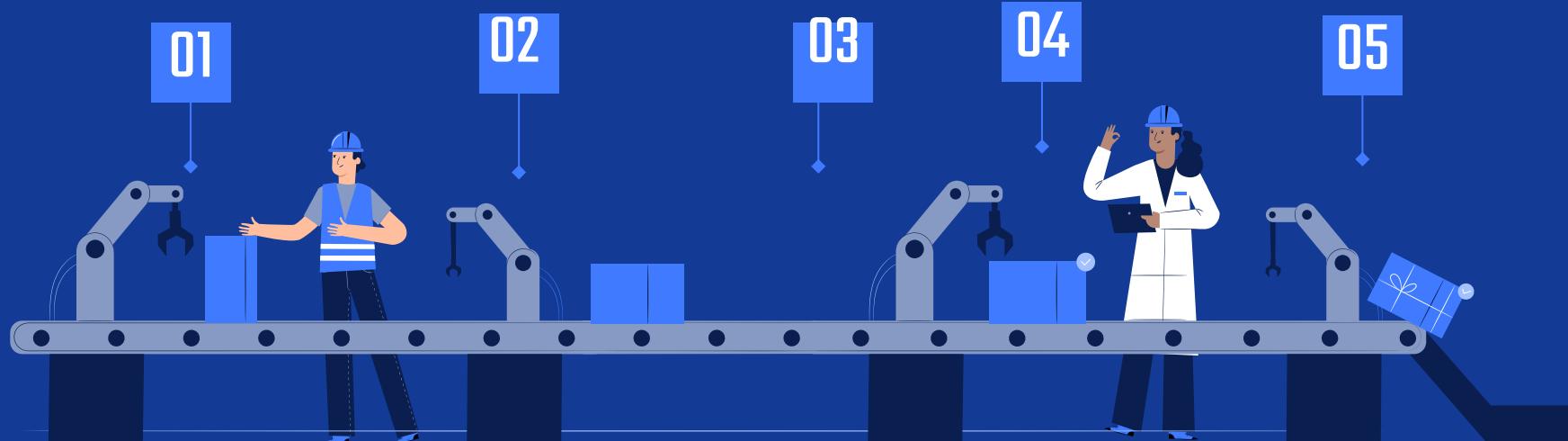
Reprises des fonctions utiles de la question 1

Code qui trouve les points d'intersection entre les diagonales et le chemin

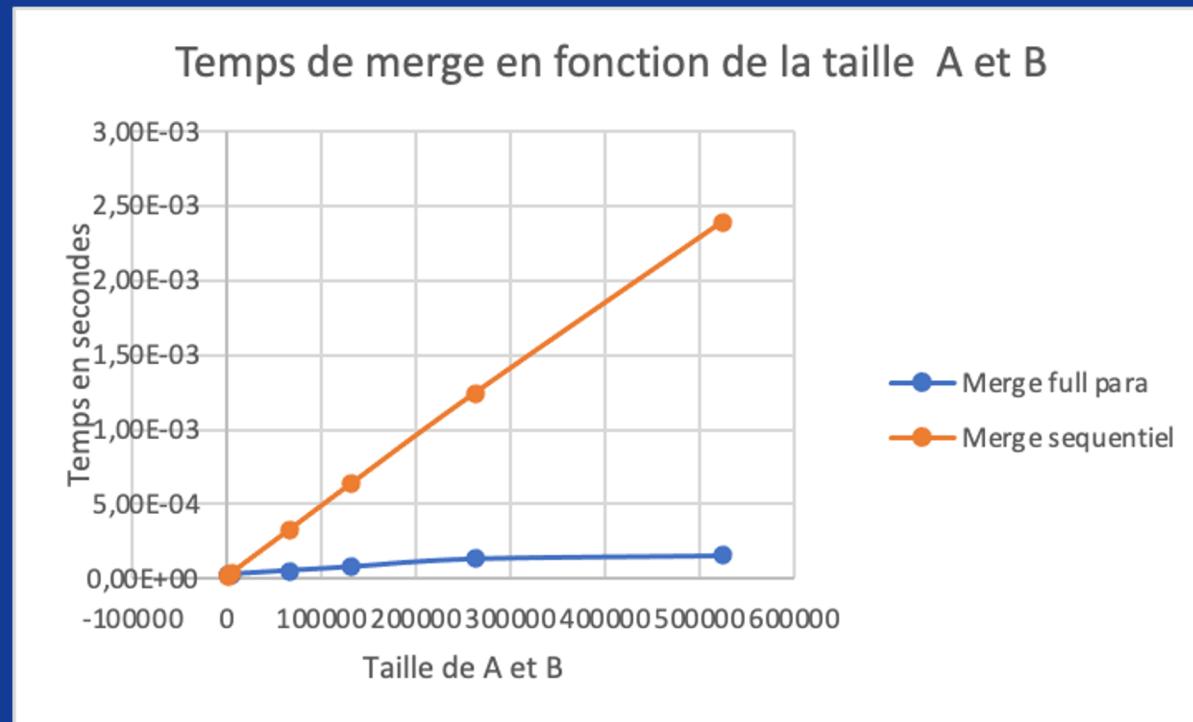
Code qui lance les fenêtres de tri 1 par 1

Code qui lance les fenêtres de tri en parallèle

Tests et mesures



Résultats des mesures : Durée globale



04

Etape 3 :
merge_sort

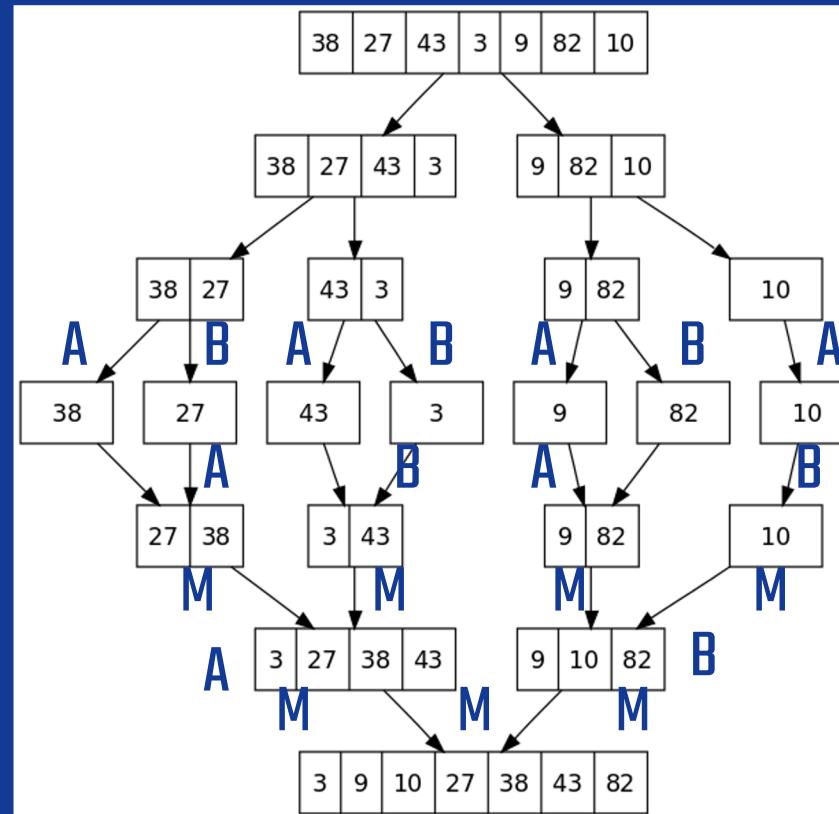


ETAPE 3



En bouclant sur les appels appropriés de `pathBig_k` et de `mergeBig_k`, écrivez une fonction qui trie tout tableau M de taille d suffisamment plus petite que la mémoire globale. Donnez le temps d'exécution par rapport à d .

Merge Sort



Logique du code

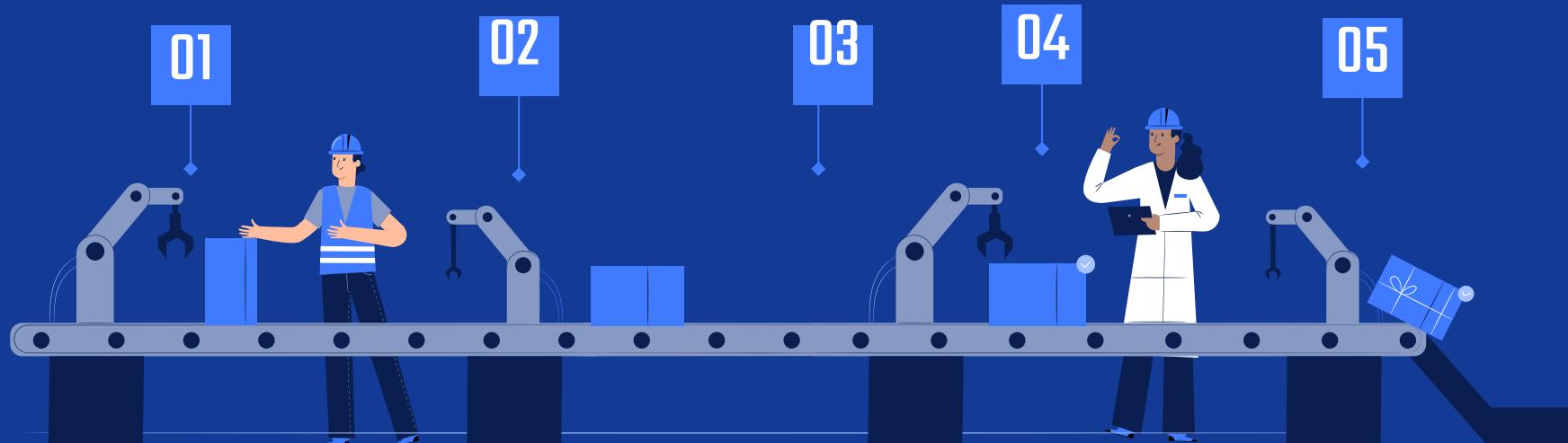
Code en séquentiel en C

Reprises des fonctions utiles de la question 1 et 2

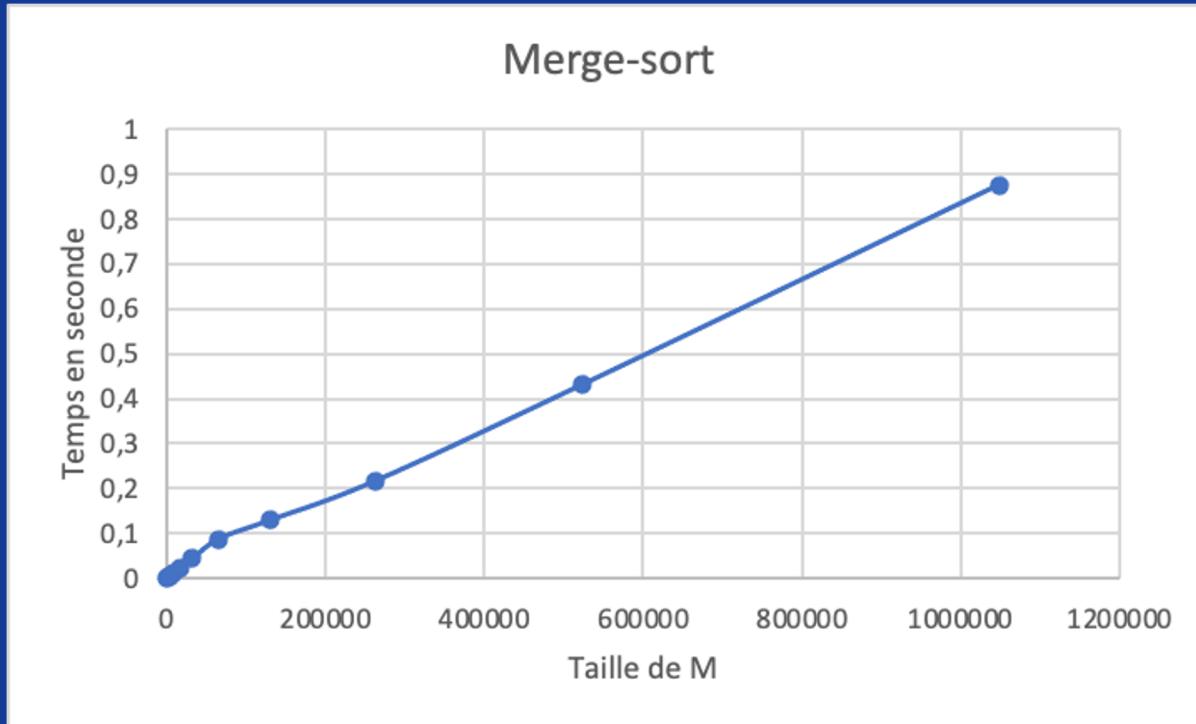
“ Conversion ” du code séquentiel en CUDA

Tests et mesures

Amélioration du code

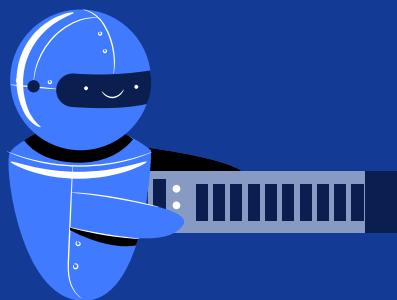


Résultats des mesures



05

Etape 5 : `mergeSmallBatch_k`



ETAPE 4

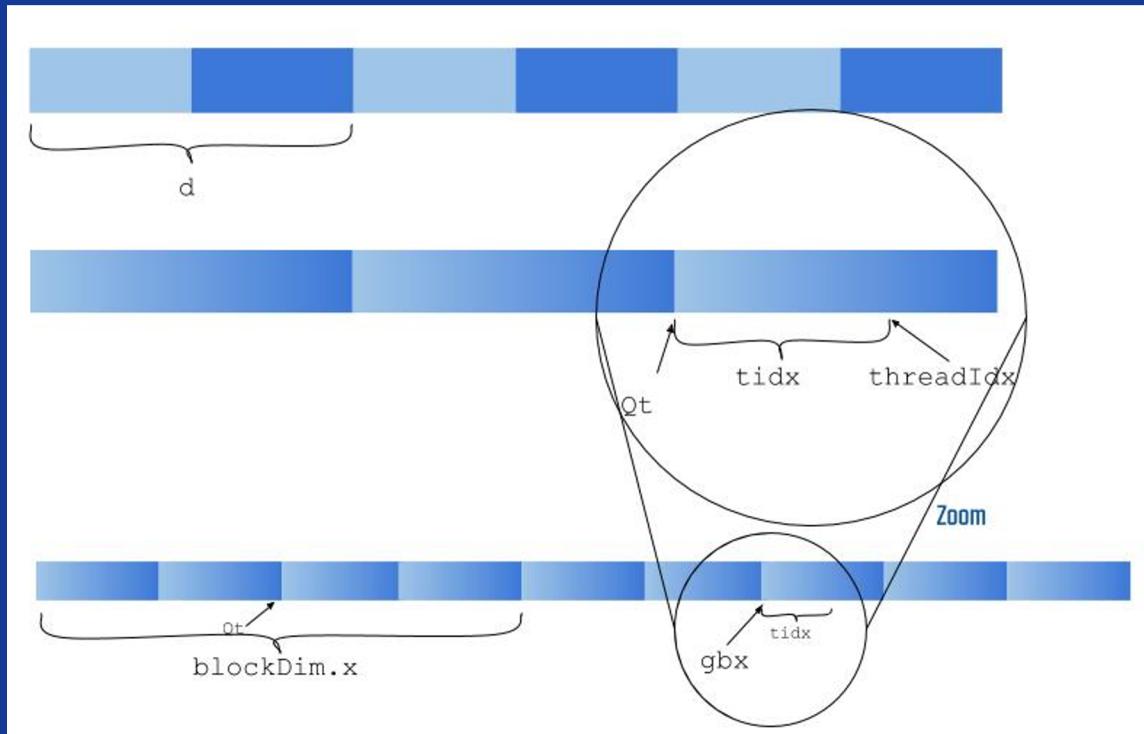


Expliquer pourquoi les indices

- `int tidx = threadIdx.x%d;`
- `int Qt = (threadIdx.x-tidx)/d;`
- `int gbx = Qt +
blockIdx.x*(blockDim.x/d);`

Sont importants dans la définition de
`mergeSmallBatch k`.

Question 4



ETAPE 5



Écrivez le noyau
mergeSmallBatch k qui
fusionne deux par deux
 $\{A_i\}_{1 \leq i \leq N}$ et $\{B_i\}_{1 \leq i \leq N}$

Donnez le temps d'exécution
par rapport à $d = 4, 8, \dots, 1024$.

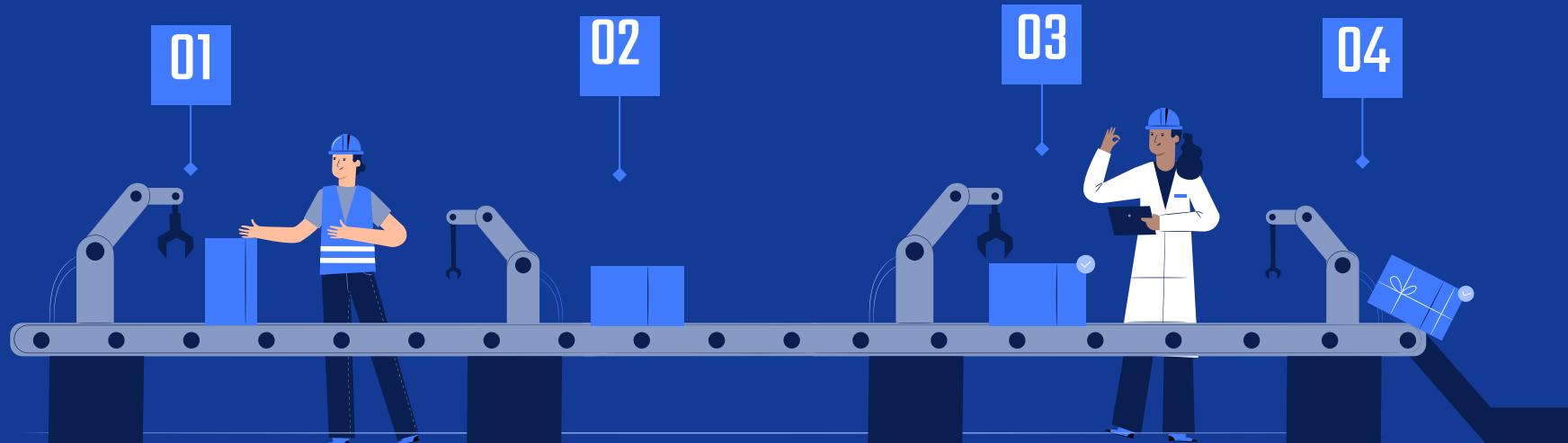
Logique

Reprise de la fonction
MergeSmall

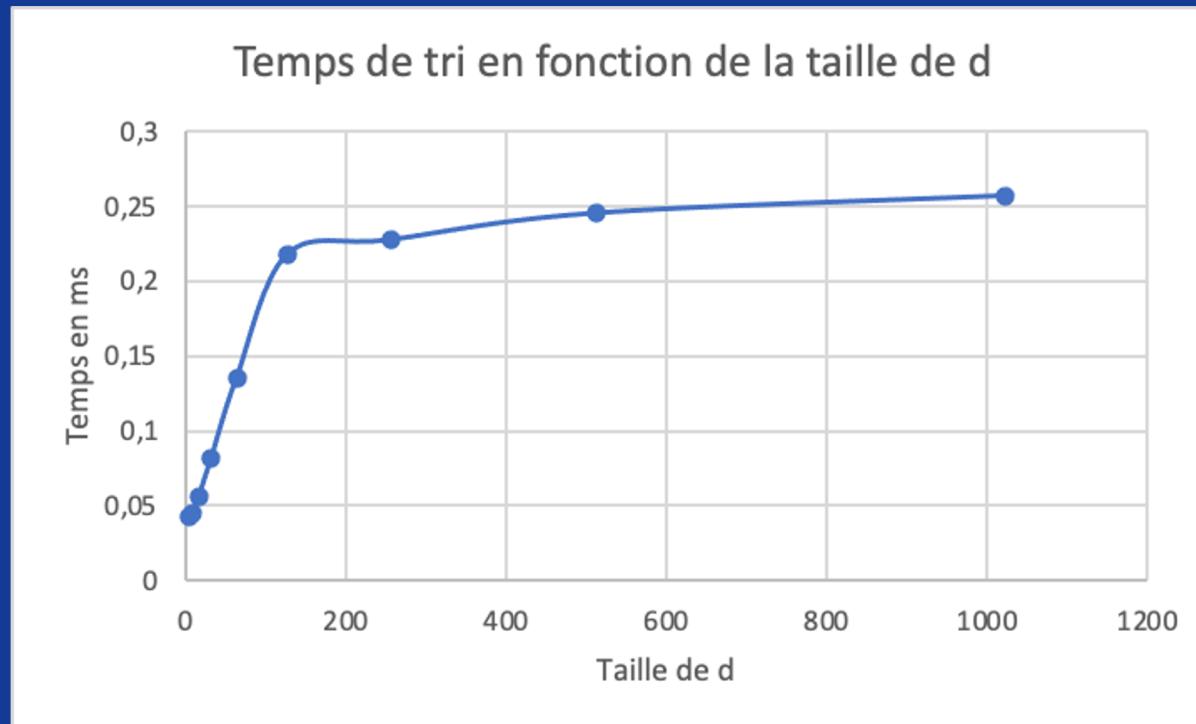
Utilisation des indices
de la question 4

Tests et
mesures

Optimisation du
code

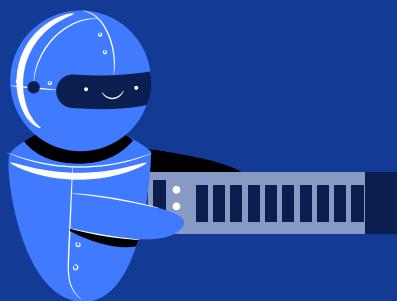


Résultats des mesures



05

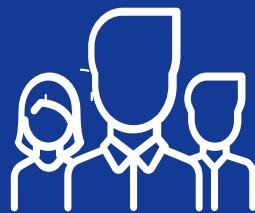
Applications



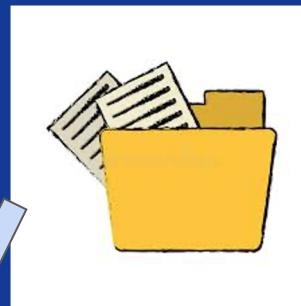
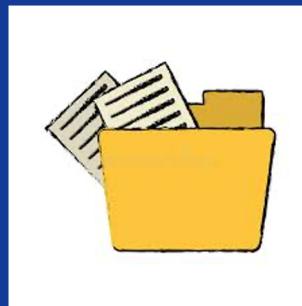


D'un point de vue
“pratique”

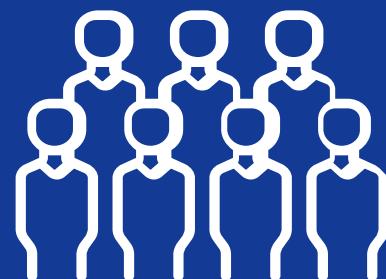
Fusion d'entreprise



Entreprise A

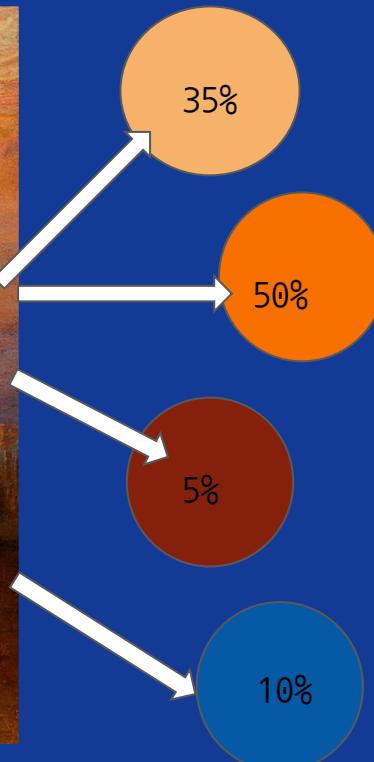
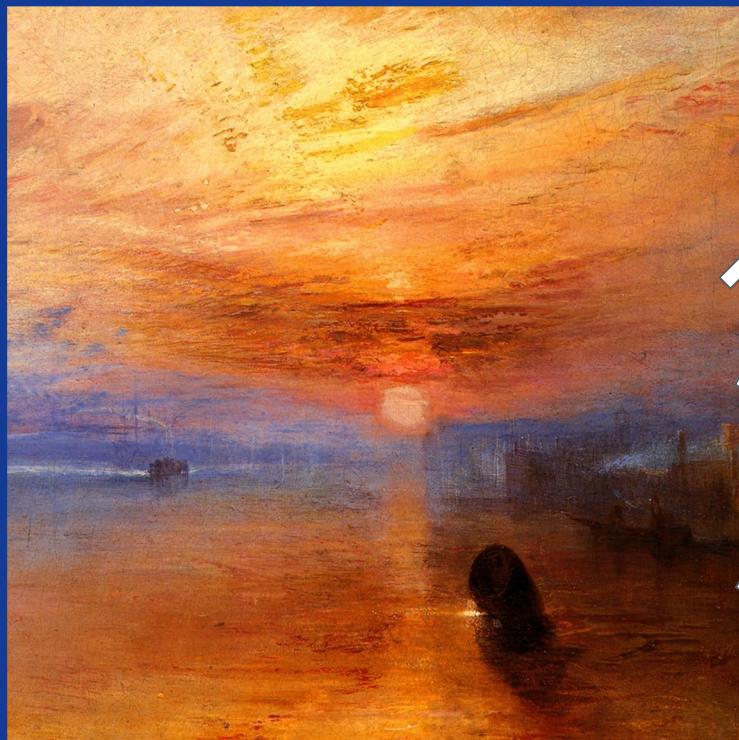


Entreprise B



Entreprise M

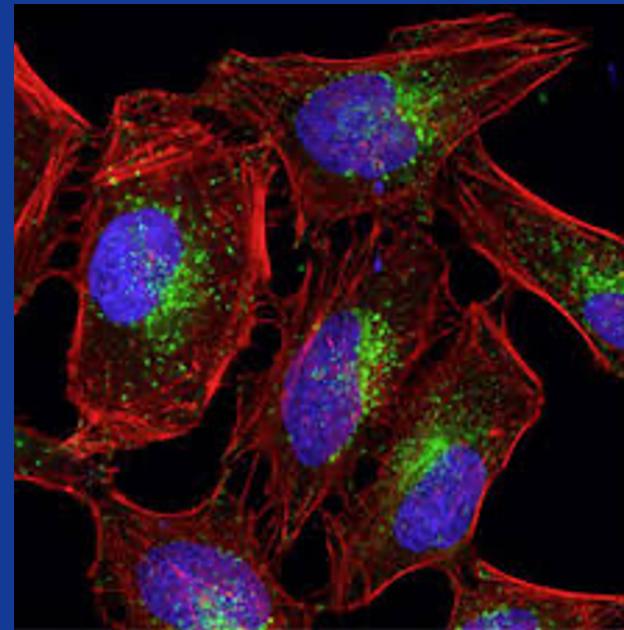
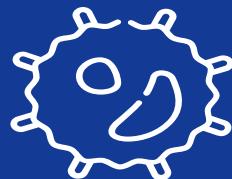
Traitements images : dans l'art



- Reconnaître la palette de couleur
- Reconnaître le peintre
- Classer par couleurs
- Reconnaître le mouvement artistique
- Reconnaître la période de l ' artiste

Traitements images : dans la médecine

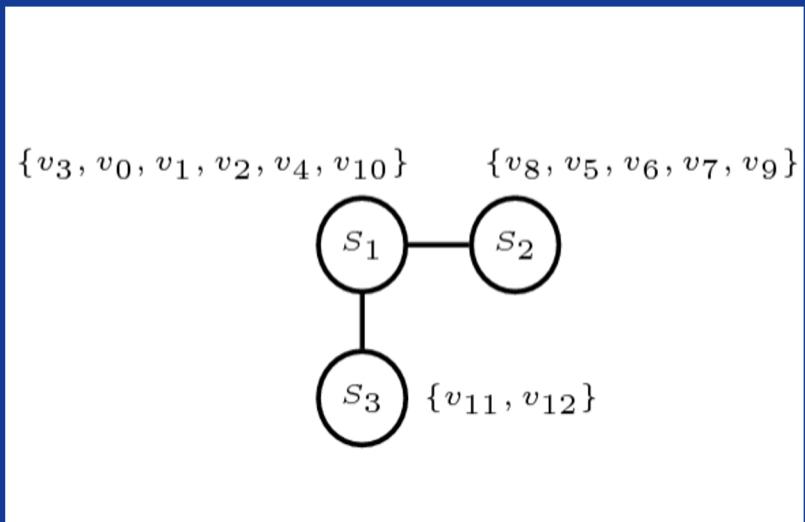
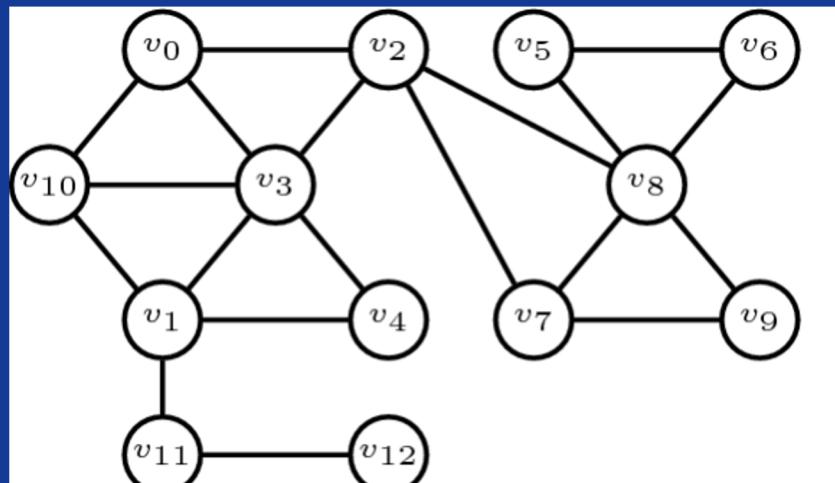
- Comptage de cellules cancéreuses
- Repérage de cellules cancéreuses



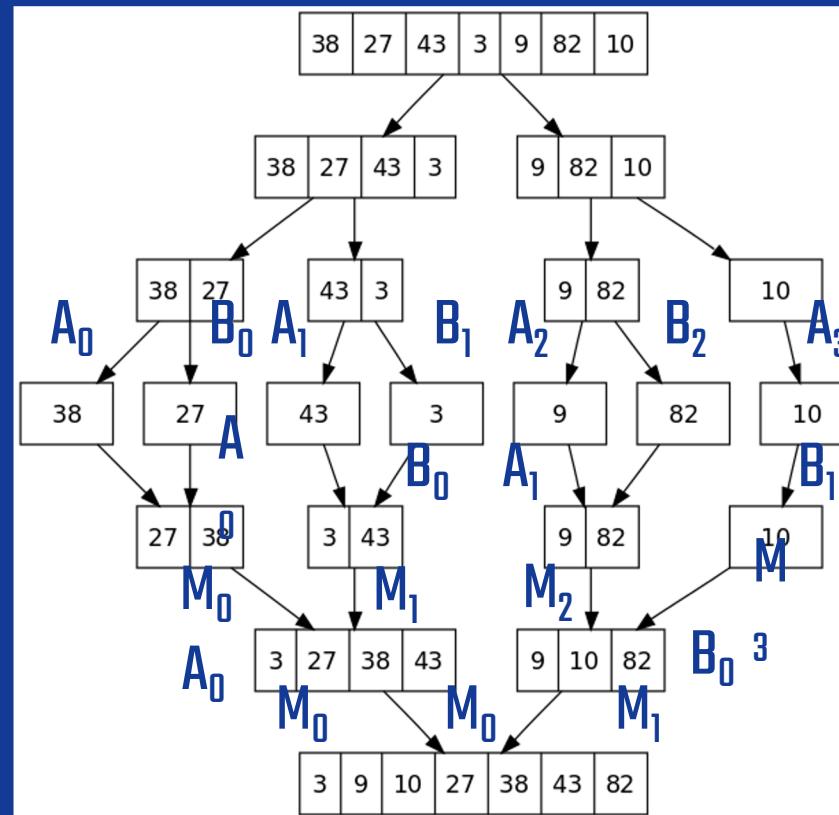


D'un point de vue
“technique”

Compression de graphes



Merge Sort avec MergeSmallBatch



COMPARAISONS

01.

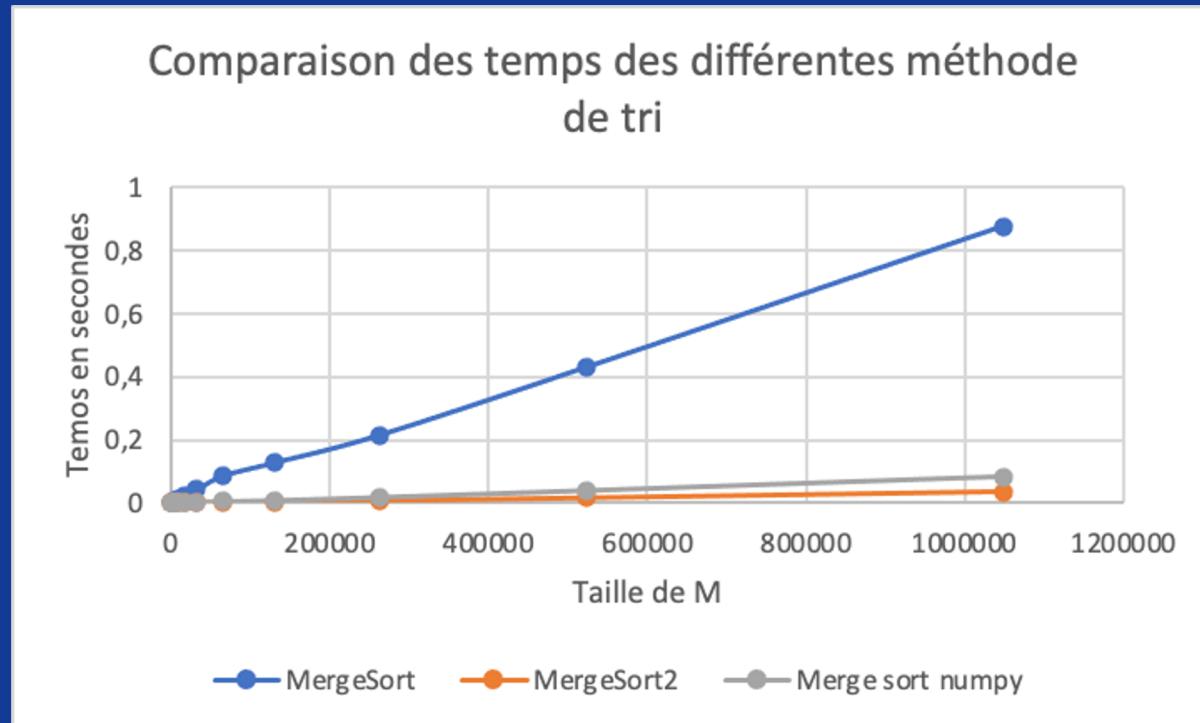
Comparaison entre
les temps de la
question 3 et 5



02.

Comparaison avec
les temps de
mergesort numpy

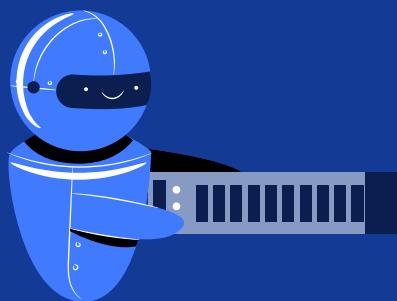
Résultats des mesures





Fin

Conclusion



Conclusion



GPU

Découvrir le monde du GPU en pratique

Cuda - Parallélisme

Découvrir un autre moyen de mettre en place du parallélisme

Difficultées

- Milieu
- Boucle infinie
- Gestion des restes

Réfléchir à différentes applications possibles



Merci

Avez-vous des
questions ?

CREDITS: This presentation template was created by Slidesgo,
including icons by Flaticon, infographics & images by Freepik and
illustrations by Stories

