

Année : 2023
 Formation : S34
 Examen : PRAM
 N° étudiant : 2190 3454

Note de l'épreuve

10.75

(1) Le Candidat doit inscrire ici : ses nom, prénoms, lieu et date de naissance, puis rabattre suivant le pointillé le coin de la copie et coller.

Il est interdit au Candidat de signer sa copie ou d'y inscrire un signe quelconque pouvant en indiquer la provenance.

(1) Nom : LATAPIE Florian
 Prénoms : Florian
 Né(e) à :
 le 27/09/2001

, le

Exercice 1

① la principale de ce tri est de comparer chaque valeur à toutes les autres dans la matrice ">" la valeur la plus grande est de somme nulle -1 et la plus petite de somme 0.
 en effet la valeur la plus petite est inférieure ou égale à toutes les autres donc somme de zéros et inversement pour la plus grande.

le tableau "+" indique la position triée de chaque élément du tableau

il suffit de mettre à l'indice + dans le résultat la valeur du tableau d'origine: $res[i] = L[+[i]]$

② il y a 18 comparaisons différentes et indépendantes pour 4 valeurs: pour n valeurs il faut n^2 processeurs

le résultat de la comparaison est lui aussi indépendant des autres

on peut donc utiliser ^{une PRAM} CRCW (lecture et écriture concurrentes)

③ le temps de chaque opération est :

$$\begin{array}{c|c|c} > & + & \downarrow \\ \hline 1 & \log(n) & 1 \end{array} = O(1 \times \log(n) \times 1) = O(\log(n))$$

④ supposons que les estimations de la question ③ sont correctes:

$$\begin{aligned} \text{work} &= \text{calcul PRAM} \times \text{nb Proc} \\ &= 1 \times \log(n) \times 1 \times n^2 \\ &= n^2 \times \log(n) \end{aligned}$$

⑤ ce n'est optimal que si $\text{work} \leq T_{\text{seq}}()$, selon mes calculs ce n'est pas le cas

2/6

Exercice 2

- ① en utilisant préfixe sur un tableau de flags pour noter uniquement les éléments intéressants on peut faire comme cela :

A	3	4	2	1	4	2	5	6
Flags	1	0	0	1	0	0	1	0

1.5/1.5

Sum-prefix	1	1	1	2	2	2	3	3
res	3	1	5					

on remarque qu'avec cet algorithme les valeurs impaires sont bien à gauche du tableau final

- ② le nombre d'éléments intéressants total est stocké dans la dernière case du tableau sum-prefix, dans notre exemple : 3

nous n'avons pas forcément besoin de stocker et incrémenter une autre variable globale

Si l'on souhaite une ER PRAM on peut faire une boucle pour fournir l'information à tous les processeurs après l'étape sum-prefix.

cette information sera stockée dans une variable "nb_intéressants" comme indiqué dans l'énoncé

- ③ afin de connaître la position relative des non-intéressants on peut ajouter les étapes de leur calcul après sum prefix

exemple :

3/6

A	3	4	2	1	4	2	5	6
Flags	1	0	0	1	0	0	1	0
sum-pref	1	1	1	2	2	2	3	3
\neg Flags	0	1	1	0	1	1	0	1
rel-sum-pref	0	1	2	2	3	4	4	5

0.75/1

on peut donc remplir res avec
 $res[i] = A[sum_pref[i]]$ où $flags == 1$
 puis
 $res[i + nb_interessés] = A[rel_sum_pref[i]]$
 où $\neg flags == 1$

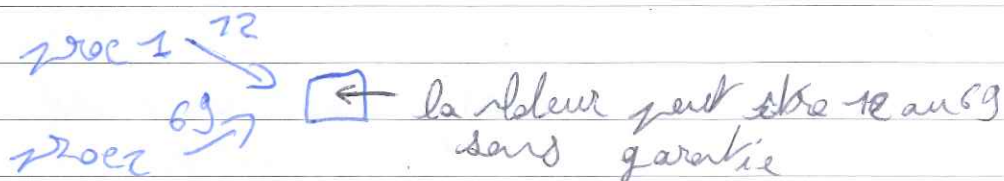
1/2

$i < len(A)$
 i dans $1, n$
 ④ for i in parallel do
 if $flags == 1$;
 $res[i] = A[sum_pref[i]]$ non, il faut lire $A(i)$
 else if $\neg flags == 1$;
 $res[i + nb_interessés] = A[rel_sum_pref$
 $[i]]$ non

Exercice 3

1.5/5

① le mode CW d'une PRAM permet à chaque processeur d'une PRAM d'écrire où il veut peu importe où en sont les autres processeurs, si aucune variante n'est précisée ils peuvent écrire des valeurs différentes dans la même adresse :

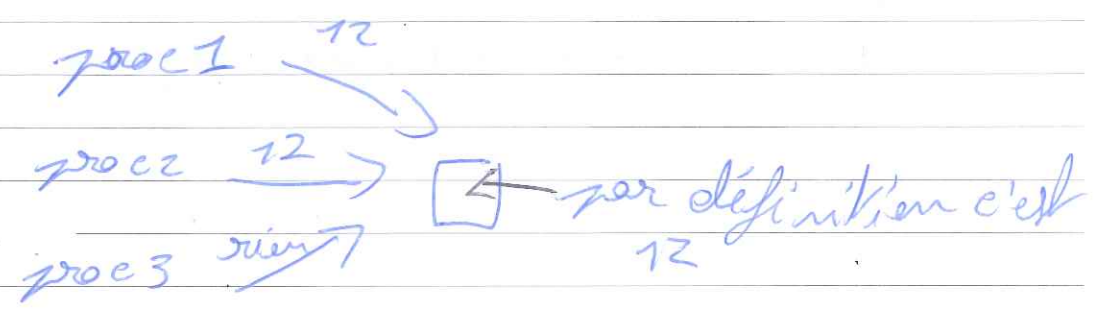


④ A/6

en variante "consistent mode" il est défini que tous les processeurs qui vont écrire à cette adresse écrira

la même valeur.

0.5



1

② l'indice du processeur définit quelle case il va remplir. il obtient cette information depuis l'adresse mémoire indiquée dans le tuple

③

Exerciel 4

1/5

$$\textcircled{1} a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7$$

