TD1, SI4 Parallel programming 2022-2023

Exercice 1

In the course, we have shown 2 versions of the parallel computation of the maximum of n values.
Q1) Explain why the proposed CRCW PRAM algorithm that is using n^2 processors can indeed compare all needed values together in just one single phase. Remember that to find the maximum of a set of n values, each of them must at some point be compared to all the others.

Why in the sequential algorithm that you can easily write down, the time complexity ends up being in O(n) (and not O(n^2)).

Q2) Explain why the proposed CRCW PRAM algorithm has to allow CR ?

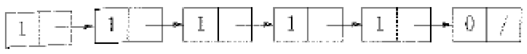
Q3) Explain why the proposed CRCW PRAM algorithm has to allow Arbitrary CW ?

Q4) Sketch how to simulate a C.R. of this algorithm, on an E.R PRAM. For the C.R simulation, write down the complete algorithm, assuming the value to be copied to the n processors, is stored in an array of size n, at index 0. Assume that n=2^m, so, you can iterate in O(m) parallel steps. Highlight why the simulation has only O(log n) parallel time complexity, given the number of data is n.

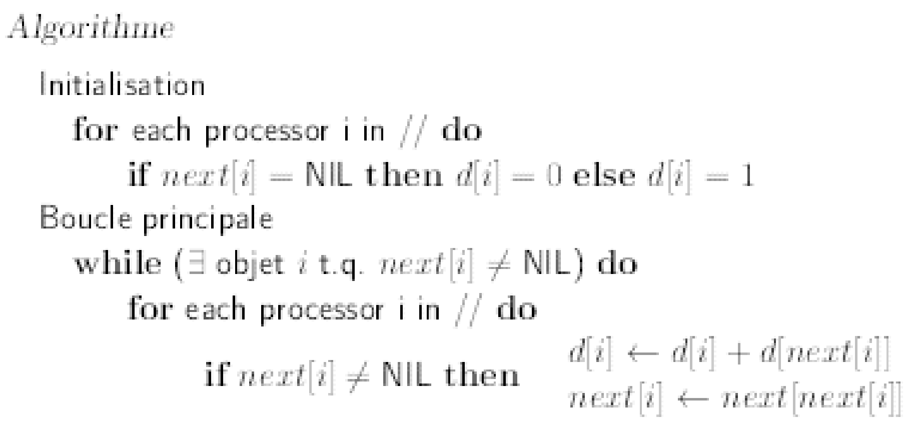
Q5) Sketch how to do the same for the Arbitrary C.W, on an E.W. PRAM.

Exercice 2

Q1) What does the following algorithm applied to a chained linked list of elements compute?

Start with this list once initialized 

Q2) What is its parallel time complexity on a PRAM (considering you can use the most powerful PRAM you need) ? Hint : how many times is the condition of the while loop executed ?
Q3) Which PRAM variant is needed at least, not to increase the parallel time complexity ? Hint: is it possible that 2 processes read data of the same list item at the same PRAM instruction ? (consider an instruction, eg, an addition with two operands as being one single instruction, ie., do not decompose even more one such operation, like an addition, into its corresponding assembly code)



```
Algorithme
  Initialisation
    for each processor i in // do
      if $next[i] = \text{NIL}$ then $d[i] = 0$ else $d[i] = 1$
  Boucle principale
    while ($\exists$ objet $i$ t.q. $next[i] \neq \text{NIL}$) do
      for each processor i in // do
        if $next[i] \neq \text{NIL}$ then
          $d[i] \leftarrow d[i] + d[next[i]]$
          $next[i] \leftarrow next[next[i]]$
```

Exercice 3

In the algorithm provided for the parallel computation of the maximum (version 2), the course has shown a classical way to derive a work optimal PRAM algorithm.
Write down, using the pseudo PRAM language, the proposed work optimal algorithm.