

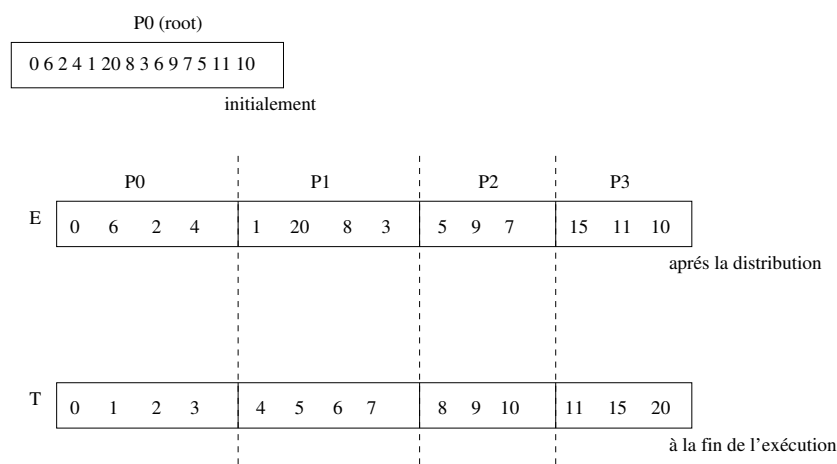
Série de Travaux Dirigés : 2 - MPI - Suite et tri par induction

Exercice 1. Parallélisation du tri par induction

Soit l'algorithme suivant qui réalise un tri appelé tri par induction

```
for i from 1 to n do
  p = 1
  for j from 1 to n do
    if (E(j)<E(i)) then
      p = p + 1
    end if
  end for
  T(p) = E(i)
end for
```

Soit E un tableau d'entiers tous distincts, on souhaite paralléliser la première boucle for (indice i) en répartissant le calcul des positions p sur les processeurs. Initialement le tableau E est généré par un processeur **root** et distribué aux autres. A la fin du tri, on souhaite que le tableau résultat reste réparti sur les **nprocs** processeurs.



1. Définissez les étapes de la parallélisation sachant que le tableau E est initialement généré le processeur **root**.
2. Par rapport à ces étapes, explicitez les fonctions de communications MPI que vous allez utiliser.
3. A partir du squelette disponible sous Celene complétez l'implémentation.

Exercice 2. Suite de Syracuse

Une suite de Syracuse est une suite telle que $U_0 = x$ avec $x > 0$ et

$$U_i = \begin{cases} \frac{U_{i-1}}{2} & \text{si } U_{i-1} \text{ est pair} \\ 3U_{i-1} + 1 & \text{sinon} \end{cases}$$

On souhaite vérifier si un tableau d'entiers de taille n correspond à une suite de Syracuse alors que ce tableau est initialement sur le processeur **root**.

1. Décrivez la parallélisation que vous envisagez sachant qu'on souhaite le cas général lorsque n n'est pas divisible par **root**.
2. Explicitez les fonctions de communications MPI que vous allez utiliser et pourquoi.
3. Proposez une implémentation à partir du squelette disponible sous Celene.