

► Exercice 1. Calcul de $n!$

On considère l'algorithme suivant

Entrée: $n > 0$
Sortie: $n!$

```
res <- 1
n2 <- n
tant que n2 > 0 faire
  res <- res * n2
  n2 <- n2 - 1
retourner res
```

1. Montrer que l'invariant $\text{res} = \prod_{i=n2+1}^n i$ est vrai au départ de la boucle et conservé ensuite. En déduire que le programme retourne $n! = \prod_{i=1}^n i$.
2. Quel est la complexité de cet algorithme?

► Exercice 2. Calcul de $\lfloor \sqrt{n} \rfloor$

On considère l'algorithme suivant où toutes les variables contiennent des entiers.

Entrée: un entier $n > 1$
Sortie: la racine de n

```
inf <- 1; sup <- n
tant que inf < sup-1 faire
  mid <- (inf + sup) / 2
  si mid*mid <= n faire
    inf <- mid
  sinon
    sup <- mid
retourner inf
```

1. Montrer que l'invariant $\text{inf}^2 \leq n < \text{sup}^2$ est vrai au départ de la boucle et conservé ensuite.
2. En déduire que si le programme termine; il retourne $\lfloor \sqrt{n} \rfloor$.
3. Montrer que la différence $\text{sup} - \text{inf}$ diminue strictement à chaque étape de boucle. En déduire que le programme termine.
4. Quelle est la complexité?

► **Exercice 3. (Calcul des deux maximums d'un tableau).**

L'algorithme suivant retourne le plus grand élément et le deuxième plus grand élément d'un tableau ainsi que leur position.

Entrée : un tableau T de taille $n > 2$.

Sortie : max , pos , max2 , pos2

```
si  $T[0] > T[1]$  alors
     $\text{max} \leftarrow T[0]$ ;  $\text{pos} \leftarrow 0$ 
     $\text{max2} \leftarrow T[1]$ ;  $\text{pos2} \leftarrow 1$ 
sinon
     $\text{max} \leftarrow T[1]$ ;  $\text{pos} \leftarrow 1$ 
     $\text{max2} \leftarrow T[0]$ ;  $\text{pos2} \leftarrow 0$ 
 $i = 2$ 
while  $i < n-1$  faire
    si  $T[i] > \text{max}$  alors
         $\text{max2} \leftarrow \text{max}$ ;  $\text{pos2} \leftarrow \text{pos}$ 
         $\text{max} \leftarrow T[i]$ ;  $\text{pos} \leftarrow i$ 
    sinon si  $T[i] > \text{max2}$  alors
         $\text{max2} \leftarrow T[i]$ ;  $\text{pos2} \leftarrow i$ 
     $i = i + 1$ 
retourner  $\text{max}$ ,  $\text{pos}$ ,  $\text{max2}$ ,  $\text{pos2}$ 
```

On veut montrer la spécification suivante :

$$\begin{aligned} \text{pos} \neq \text{pos2}, \quad T[\text{pos}] = \text{max}, \quad T[\text{pos2}] = \text{max2}, \quad \text{max2} \leq \text{max}, \\ \text{et pour tout } i \neq \text{pos}, \text{pos2} \text{ on a } T[i] \leq \text{max2} \end{aligned}$$

Pour ceci on utilise l'invariant suivant :

$$\begin{aligned} \text{pos} \neq \text{pos2}, \quad T[\text{pos}] = \text{max}, \quad T[\text{pos2}] = \text{max2}, \quad \text{max2} \leq \text{max}, \\ \text{et pour tout } j < i, \text{ si } j \neq \text{pos}, \text{pos2} \text{ alors } T[j] \leq \text{max2} \end{aligned}$$

1. Montrer que l'invariant est vérifié avant l'entrée dans la boucle while ;
2. Montrer que si l'invariant est vérifié au début d'une étape de boucle, alors il est encore vérifié à la fin ; comparaisons.
3. En déduire que l'algorithme vérifie bien la spécification.
4. Quel est la complexité de cet algorithme ?