

Exercice 1 : Correction du tri rapide

Le but de cet exercice est de démontrer la correction de l'algorithme de tri rapide tel qu'il est rappelé ci-dessous.

Algorithme 1 : Partitionner(T, d, f, p)

Entrée : Un tableau T indicé par $\llbracket 0, n \rrbracket$, trois entiers d, f et p tels que $0 \leq d \leq p \leq f < n$

Sortie : Un indice $q \in \llbracket d, f \rrbracket$ tel que $T[q]$ vaut finalement la valeur qu'avait $T[p]$ avant l'appel

Effet : Le sous-tableau $T\llbracket d, f \rrbracket$ est permuté de sorte que $\begin{cases} \forall k \in \llbracket d, q \rrbracket, T[k] \leq T[q] \\ \forall k \in \llbracket q, f \rrbracket, T[k] > T[q] \end{cases}$

```
1 pivot ← T[p];
2 échanger T[p] et T[d]; // on place le pivot au début
3 a ← d + 1;
4 b ← f;
5 tant que a ≤ b faire
6   si T[a] ≤ pivot alors
7     a ← a + 1
8   sinon
9     échanger T[a] et T[b];
10    b ← b - 1;
11 échanger T[d] et T[a - 1];
12 retourner a - 1
```

Algorithme 2 : Procédure tri_rapide(T, d, f)

Entrée : Un tableau T indicé par $\llbracket 0, n \rrbracket$, deux entiers $d \leq 0$ et $f \leq n - 1$

Sortie : Aucune sortie, mais $T\llbracket d, f \rrbracket$ est triée par ordre croissant

```
1 si d < f alors
2   p ← choix_pivot(T, d, f) // choix d'un entier entre d et f;
3   q ← Partitionner(T, d, f, p);
4   tri_rapide(T, d, q - 1);
5   tri_rapide(T, q + 1, f)
```

- Q. 1 Exécuter l'algorithme Partitionner sur le tableau $T : [0; 1; 5; 3; 4; 2]$ pour $d = 2$ et $f = 5$. Représenter l'état du tableau T à chaque étape ainsi que les valeurs des variables a et b .
- Q. 2 Proposer un invariant ♣ de boucle pour Partitionner qui précise les valeurs possibles et surtout le rôle des variables a et b .
- Q. 3 Démontrer que l'invariant proposé est bien un invariant de boucle.
- Q. 4 On admet que tout au long de l'algorithme le tableau T est une permutation du tableau initial, et que cette éventuelle transformation ne concerne que des indices compris dans l'intervalle $\llbracket d, f \rrbracket$ (donc la partie de T extérieure à cet intervalle n'est pas modifiée). À l'aide de l'invariant proposé à la question précédente démontrer que l'algorithme Partitionner est correct.
- Q. 5 En admettant la correction de l'algorithme Partitionner, démontrer que l'algorithme Tri_rapide est correct.

♣. celui-ci peut être composé de plusieurs propriétés