

1 Tri par insertion

On va prouver l'état final de l'algorithme du tri par insertion d'un tableau. Cet état final est exprimé par une permutation des indices, ce qui assure que les valeurs du tableau ne sont ni dupliquées ni supprimées lorsqu'elles apparaissent à de multiples exemplaires. Imaginez les conséquences de duplications ou de suppressions de valeurs si le tableau contient les comptes des clients d'une banque...

Les objectifs de cet exercice sont :

- de s'habituer à ne pas confondre variables et valeurs,
- de faire des calculs sur les indices,
- de gérer des notations multiples sans perdre le fil du raisonnement,
- de se familiariser avec l'expression d'assertions et d'invariants,
- de développer une preuve rigoureuse.

Une Action TriParInsertion(**la donnée** N : une constante entière ≥ 0 , **la donnée-résultat** T : un tableau de $1...N$ d'entiers)

Etat initial : T contient N entiers t_1, \dots, t_N

Etat final : Il existe k une permutation de $\{1, \dots, N\}$ t.q. T contient les valeurs $t_{k(1)}, \dots, t_{k(N)}$, et $\forall i, 1 \leq i \leq N-1 \Rightarrow t_{k(i)} \leq t_{k(i+1)}$

Lexique : clef, i , j : trois entiers

```

1  $j \leftarrow 2$ 
2 tant que  $j \leq N$  faire
    { $I_1$ : Il existe  $\bar{k}$  une permutation de  $\{1, \dots, j-1\}$  t.q.  $T$  contient les valeurs
       $t_{\bar{k}(1)}, \dots, t_{\bar{k}(j-1)}, t_j, \dots, t_N$ , et  $\forall i, 1 \leq i \leq j-2 \Rightarrow t_{\bar{k}(i)} \leq t_{\bar{k}(i+1)}$  }
3   Clef  $\leftarrow T[j]$ 
4    $i \leftarrow j - 1$ 
    { $A$  : Soient  $\bar{t}_1, \dots, \bar{t}_j, \dots, \bar{t}_n$  les valeurs de  $T$  }
5   tant que  $i \geq 1$  et puis  $T[i] > \text{Clef}$  faire
6      $T[i+1] \leftarrow T[i]$ 
7      $i \leftarrow i - 1$ 
    { $I_2$  :  $T$  contient les valeurs  $\bar{t}_1, \dots, \bar{t}_i, \bar{t}_{i+1}, \bar{t}_{i+1}, \dots, \bar{t}_{j-1}, \bar{t}_{j+1}, \dots, \bar{t}_n$  et  $\text{Clef} < \bar{t}_{i+1}$  }
8    $T[i+1] \leftarrow \text{Clef}$ 
9    $j \leftarrow j + 1$ 

```

Cet algorithme est composé d'une boucle principale (lignes 2 à 9) et d'une sous-boucle (lignes 5 à 7). L'assertion I_1 est un invariant de la boucle principale, placé au début de la boucle principale. L'assertion I_2 est un invariant de la sous-boucle, placé à la fin de cette sous-boucle. L'assertion A est placée avant la sous-boucle, c'est une simple renotation des valeurs, qui a pour but d'alléger les notations dans l'assertion I_2 .

Dans la suite, on commence par prouver l'invariant I_2 (section 1.1). On se sert de ce résultat pour prouver l'invariant I_1 (section 1.2). La preuve de l'état final est alors triviale (section 1.3).

1.1 Preuve de l'invariant I_2

Initialisation :

- A l'entrée de la sous-boucle, i contient la valeur de $j-1$
- Après l'action $T[i+1] \leftarrow T[i]$ on a :

<i>Indices</i>	1	...	$j-1$	j	$j+1$...	N
$T =$	\bar{t}_1	...	\bar{t}_{j-1}	\bar{t}_{j-1}	\bar{t}_{j+1}	...	\bar{t}_N

- Après l'action $i \leftarrow i - 1$, i contient la valeur de $j-2$, donc $i+1$ vaut la valeur de $j-1$ et ainsi :

<i>Indices</i>	1	...	$j-1$	j	$j+1$...	N
$T =$	\bar{t}_1	...	\bar{t}_{i+1}	\bar{t}_{i+1}	\bar{t}_{j+1}	...	\bar{t}_N

L'assertion I_2 est donc vérifiée à la fin de la première itération.

Conservation :

- On suppose l'invariant I_2 vérifié lorsque i vaut i_0 .
- On suppose donc que T contient

<i>Indices</i>	1	...	i_0-1	i_0	i_0+1	i_0+2	...	j	$j+1$...	N
$T =$	\bar{t}_1	...	\bar{t}_{i_0-1}	\bar{t}_{i_0}	\bar{t}_{i_0+1}	\bar{t}_{i_0+1}	...	\bar{t}_{j-1}	\bar{t}_{j+1}	...	\bar{t}_N

et que $Clef < \bar{t}_{i_0+1}$

- Puisque l'on rentre dans la prochaine itération, on sait que $i_0 \geq 1$ et que $Clef < \bar{t}_{i_0}$
- Après l'action $T[i+1] \leftarrow T[i]$, on a :

<i>Indices</i>	1	...	i_0-1	i_0	i_0+1	i_0+2	...	j	$j+1$...	N
$T =$	\bar{t}_1	...	\bar{t}_{i_0-1}	\bar{t}_{i_0}	\bar{t}_{i_0}	\bar{t}_{i_0+1}	...	\bar{t}_{j-1}	\bar{t}_{j+1}	...	\bar{t}_N

- Après l'action $i \leftarrow i-1$, i contient la valeur i_0-1 , donc :

<i>Indices</i>	1	...	i_0-1	i_0	i_0+1	i_0+2	...	j	$j+1$...	N
$T =$	\bar{t}_1	...	\bar{t}_i	\bar{t}_{i+1}	\bar{t}_{i+1}	\bar{t}_{i+2}	...	\bar{t}_{j-1}	\bar{t}_{j+1}	...	\bar{t}_N

Comme par ailleurs $Clef < \bar{t}_{i_0} = \bar{t}_{i+1}$, l'assertion I_2 est vérifiée à la fin du corps l'itération suivante. On a donc prouvé que si l'assertion I_2 est vérifiée à une itération, alors elle est vérifiée à l'itération suivante. Par conséquent, I_2 est un invariant de l'itération.

1.2 Preuve de l'invariant I_1

Initialisation :

- À la première itération, j vaut 2. Donc $\{1, \dots, j-1\} = \{1\}$, $\bar{k}(1) = 1$, et puisqu'il n'existe aucun i t.q $1 \leq i \leq j-2$, l'assertion I_1 est vraie.

Conservation :

- On suppose I_1 vérifiée lorsque j vaut j_0 .
- On connaît donc \bar{k} une permutation de $\{1, \dots, j_0-1\}$ t.q. T contient les valeurs $t_{\bar{k}(1)}, \dots, t_{\bar{k}(j_0-1)}, t_{j_0}, \dots, t_N$, et $\forall i, 1 \leq i \leq j_0-2 \Rightarrow t_{\bar{k}(i)} \leq t_{\bar{k}(i+1)}$
- l'exécution de la ligne 3 assure que $Clef$ vaut t_{j_0} .
- l'assertion A renote les valeurs de T de la manière suivante : $\bar{t}_1 = t_{\bar{k}(1)}, \dots, \bar{t}_{j_0-1} = t_{\bar{k}(j_0-1)}, \bar{t}_{j_0} = t_{j_0}, \dots, \bar{t}_N = t_N$
- L'invariant I_2 nous assure que à la dernière itération de la sous-boucle,

<i>Indices</i>	1	...	i_0-1	i_0	i_0+1	i_0+2	...	j	$j+1$...	N
$T =$	$t_{\bar{k}(1)}$...	$t_{\bar{k}(i_0-1)}$	$t_{\bar{k}(i_0)}$	$t_{\bar{k}(i_0+1)}$	$t_{\bar{k}(i_0+1)}$...	$t_{\bar{k}(j_0-1)}$	t_{j_0+1}	...	t_N

et $Clef < t_{\bar{k}(i_0+1)}$

- Puisque l'on sort de la sous-boucle, on doit avoir $i_0 = 0$ ou alors $\{i_0 \geq 1 \text{ et } Clef \geq t_{\bar{k}(i_0)}\}$
- Si $\{i_0 \geq 1 \text{ et } Clef \geq t_{\bar{k}(i_0)}\}$
→ l'exécution de la ligne 8 assure que :

<i>Indices</i>	1	...	i_0-1	i_0	i_0+1	i_0+2	...	j	$j+1$...	N
$T =$	$t_{\bar{k}(1)}$...	$t_{\bar{k}(i_0-1)}$	$t_{\bar{k}(i_0)}$	$Clef$	$t_{\bar{k}(i_0+1)}$...	$t_{\bar{k}(j_0-1)}$	t_{j_0+1}	...	t_N

→ l'exécution de la ligne 9 assure que j vaut $j_0 + 1$ et :

<i>Indices</i>	1	...	$i_0 - 1$	i_0	$i_0 + 1$	$i_0 + 2$...	$j - 1$	j	...	N
$T =$	$t_{\bar{k}(1)}$...	$t_{\bar{k}(i_0-1)}$	$t_{\bar{k}(i_0)}$	<i>Clef</i>	$t_{\bar{k}(i_0+1)}$...	$t_{\bar{k}(j-2)}$	t_j	...	t_N

→ On définit \tilde{k} une permutation de $\{1, \dots, j_0\} = \{1, \dots, j - 1\}$ par $\tilde{k}(1) = \bar{k}(1), \dots, \tilde{k}(i_0) = \bar{k}(i_0), \tilde{k}(i_0 + 1) = j_0, \tilde{k}(i_0 + 2) = \bar{k}(i_0 + 1), \dots, \tilde{k}(j_0) = \bar{k}(j_0 - 1)$ (informellement \tilde{k} résulte de l'insertion dans \bar{k} de j_0 à la place $i_0 + 1$). Avec cette définition de \tilde{k} , T contient :

<i>Indices</i>	1	...	$i_0 - 1$	i_0	$i_0 + 1$	$i_0 + 2$...	$j - 1$	j	...	N
$T =$	$t_{\tilde{k}(1)}$...	$t_{\tilde{k}(i_0-1)}$	$t_{\tilde{k}(i_0)}$	$t_{\tilde{k}(i_0+1)}$	$t_{\tilde{k}(i_0+2)}$...	$t_{\tilde{k}(j-1)}$	t_j	...	t_N

et les propriétés $Clef < t_{\bar{k}(i_0+1)}$ et $Clef \geq t_{\bar{k}(i_0)}$ assurent que l'invariant I_2 est vérifié à l'itération suivante, i.e. pour j valant $j_0 + 1$.

— Si $i_0 = 0$

→ on sait par l'invariant I_2 que $Clef < t_{\bar{k}(1)}$

→ L'exécution des lignes 8 et 9 assure que :

<i>Indices</i>	1	2	...	$j - 1$	j	...	N
$T =$	<i>Clef</i>	$t_{\bar{k}(1)}$...	$t_{\bar{k}(j-2)}$	t_j	...	t_N

→ On définit \tilde{k} une permutation de $\{1, \dots, j_0\} = \{1, \dots, j - 1\}$ par $\tilde{k}(1) = j_0, \tilde{k}(2) = \bar{k}(1), \dots, \tilde{k}(j_0) = \bar{k}(j_0 - 1)$ (informellement \tilde{k} résulte de l'insertion dans \bar{k} de j_0 à la place 1). Avec cette définition de \tilde{k} , T contient :

<i>Indices</i>	1	...	$j - 1$	j	...	N
$T =$	$t_{\tilde{k}(1)}$...	$t_{\tilde{k}(j-1)}$	t_j	...	t_N

et la propriété $Clef < t_{\bar{k}(1)} = t_{\bar{k}(2)}$ et assure que l'invariant I_2 est vérifié à l'itération suivante, i.e. pour j valant $j_0 + 1$.

1.3 Preuve de l'état final

Lorsque l'on sort de la boucle principale, j vaut $N + 1$, et l'invariant I_1 assure que l'état final est vérifié.