

Programmation impérative en langage C

feuille de TDM n°8 : Développement avec preuve

Exercice 1 - Calcul d'un plateau

Soit $N \in \text{INT}^*$, et $B(0..N-1)$ un tableau d'entiers croissants (non strictement).

Pour un tableau $B(0..N-1)$:

- on appelle sous-tableau tout tableau extrait $B(d, f)$ avec $0 \leq d \leq f \leq N-1$.
- On appelle plateau de B tout sous-tableau dont les cases contiennent toutes la même valeur.

On souhaite écrire un programme qui retourne dans p la longueur du plus long plateau de B .

Par exemple, avec le tableau suivant :

0	3	3	4	4	4	5	5	5	5	8	8	8	10
---	---	---	---	---	---	---	---	---	---	---	---	---	----

le plateau maximal a une longueur 4, et il commence à l'indice 6

avec la numérotation des tableaux en C.

Travail à faire

1. Ecrire la spécification de ce programme.
2. Prouver que le programme suivant est solution :

```

:
/* Q */
j = 1;
p = 1;
/* INV */
while (j < N)
{
/* INV ∧ (j < N) */
if (B[j - p] == B[j]) p = p + 1;
j = j + 1;
}
/* INV ∧ (j = N) */
/* R */

```

où :

- Q est le prédicat d'entrée
 - l'invariant est :

$$INV = (0 \leq p \leq j \leq N) \\ \wedge (\exists K : 0 \leq K \leq j - p \wedge B[K] = B[K + p - 1]) \\ \wedge (\forall K : (0 \leq K \leq j - p - 1) \rightarrow (B[K] \neq B[K + p]))$$
 - R est le prédicat de sortie
3. Écrire le programme C correspondant et le tester.

Exercice 2 - Recherche d'une valeur par dichotomie

On considère un tableau T de taille N , contenant des **entiers ordonnés** :

$$\{\forall i, j, (0 \leq i \leq j \leq N - 1 \rightarrow T[i] \leq T[j])\}$$

On veut retrouver un entier p contenu dans le tableau, en appliquant la méthode de recherche par dichotomie :

- on coupe l'intervalle de recherche $[a, b]$ en deux et à l'indice k correspondant on regarde :
 - si $T[k] = p$ on retourne k
 - si $T[k] < p$ on recommence en prenant comme intervalle de recherche $]k, b]$
 - si $p < T[k]$ on recommence en prenant comme intervalle de recherche $[a, k[$

Travail à faire

1. Spécifier les conditions d'entrée et de sortie du programme,
2. A partir du prédicat de sortie, trouver l'invariant, la condition de boucle et les initialisations.
3. Écrire le programme commenté par les prédicat et faire la preuve formelle.
4. Écrire un programme C qui effectue une recherche par dichotomie d'un entier appartenant à un tableau, en affichant l'indice k_p pour lequel $T[k_p] = p$.

Remarque : La valeur k_p à renvoyer est le plus grand indice du tableau T dont la case contient la valeur p (en effet, le tableau n'étant pas strictement croissant, on peut avoir un plateau de p de longueur quelconque).

On pourra reprendre le tri du TP 3 et en faire une fonction pour trier préalablement le tableau.

Remarque : comme on suppose que la valeur recherchée est dans le tableau (cf. prédicat d'entrée) ce point ne sera pas vérifié.

5. Modifier le programme pour qu'il effectue une recherche par dichotomie d'un entier dans un tableau, et qui, si cet entier n'appartient pas au tableau, trouve l'indice i du tableau où on devrait insérer l'entier pour le laisser ordonné ($T[i-1] < p \wedge T[i] > p$), et l'insère. Ecrire les spécifications d'entrée et de sortie de ce programme (pas la preuve).