L3-INFO Algorithmique

Quick 2 – 5 novembre 2025 – Durée 1 h

Une feuille A4 manuscrite recto-verso autorisée, ainsi que les dictionnaires bilingues.
Les dispositifs électroniques sont interdits, à l'exception des montres non connectées.
Le barème est indicatif.

Vous êtes vivement encouragés à illustrer vos recherches et vos réponses par des schémas.

Un tableau de *Young* est un tableau bidimensionnel de taille $m \times n$, tel que tous les éléments d'une même ligne sont triés de gauche à droite et tous ceux d'une même colonne sont triés de haut en bas. Si on a moins de $m \times n$ valeurs à stocker, on complète en remplissant les cellules « vides » par une valeur spéciale $+\infty$ (supérieure à toutes les autres).

Ainsi, un tableau possible pour stocker les valeurs $\{9, 16, 3, 2, 4, 8, 5, 14, 12\}$ est le tableau Y suivant :

$$Y = \begin{array}{c|c|c|c|c|}
 & 1 & \cdots & & n \\ \hline
1 & 2 & 8 & 12 & 14 \\ \hline
 & 3 & 9 & 16 & +\infty \\ \hline
\vdots & 4 & +\infty & +\infty & +\infty \\ \hline
 & 5 & +\infty & +\infty & +\infty \\ \hline
m & +\infty & +\infty & +\infty & +\infty \\ \hline
\end{array}
\quad \downarrow \leqslant$$

$$\xrightarrow{\qquad\qquad\qquad} \\ \leqslant$$

On indexe les lignes de Y de 1 à m, et ses colonnes de 1 à n.

1. **Minimum**

(1 pt) (a) Où se trouve le plus petit élément d'un tableau de Young non vide ?

(1 pt) (b) Que peut-on dire de $Y[m][n]$ si le tableau n'est pas plein ?

2. **Exemple d'insertion**

On insère le nombre 6 dans la cellule à la ligne 2 et à la colonne 4 du tableau de l'exemple.

(1 pt) Montrez comment on peut le « remonter » de proche en proche jusqu'à sa place en ne faisant que des échanges avec des cellules voisines, de façon à obtenir un tableau de Young.

3. **Percolation**

On suppose que l'on remplace la valeur de la cellule à la ligne i et à la colonne j par une valeur $v < Y[i][j]$.

(3 pts) (a) Écrivez une fonction `Remonte(i,j,v,Y)` qui « remonte » le nouvel élément placé dans la cellule $Y[i][j]$ pour le mettre à sa place.

(2 pts) (b) Donnez et justifiez que le coût de cette fonction en nombre de comparaisons avec des élements de Y.

4. **Recherche**

Voici un algorithme qui recherche si une valeur x est présente ou non dans un tableau de Young :

RECHERCHE_YOUNG(x, Y)

Données : Une valeur x et un tableau de Young Y de taille $m \times n$

Résultat : Les coordonnées d'une cellule de Y contenant x,
ou bien $(0, 0)$ si x n'est présent nulle part dans Y

1 $i := m$
2 $j := 1$
3 **tant que** $i > 0$ *et* $j \leqslant n$ **faire**
4   **si** $Y[i][j] < x$ **alors**
5         $j := j + 1$
6   **sinon si** $Y[i][j] > x$ **alors**
7         $i := i - 1$
8   **sinon**
9   **renvoyer** (i, j)
10 **renvoyer** $(0, 0)$

(2 pts) (a) Justifiez que cet algorithme se termine à l'aide d'un variant.

(b) Il est assez clair que quand l'algorithme renvoie un couple $(i, j) \neq (0, 0)$ ce résultat respecte la spécification.

Il nous reste donc à démontrer la **postcondition** suivante :

« *Quand l'algorithme renvoie* $(0, 0)$ *aucune cellule de* Y *ne contient* x. »

(1 pt) i. Que peut-on affirmer à propos de la boucle **tant que** lorsque l'algorithme a renvoyé $(0, 0)$?

(1 pt) ii. Représentez sous forme de schéma un invariant vérifié par la boucle **tant que** de notre algorithme (juste avant la ligne 4), à propos de la valeur x et du contenu de certaines cellules de Y.

En sortie de boucle, votre invariant devrait bien entendu permettre de démontrer la postcondition donnée ci-dessus.

(2 pts) iii. Exprimez également cet invariant sous la forme d'une phrase en français, en faisant référence explicitement à des indices du tableau.

(1 pt) iv. Justifiez que cet invariant est vérifié après la ligne 2 de l'algorithme.

(3 pts) v. Démontrez que la boucle **tant que** maintient effectivement cet invariant.

(2 pts) vi. Enfin, à l'aide de votre invariant en sortie de boucle, démontrez la postcondition énoncée ci-dessus.