

Feuille de Travaux Pratiques Tableau d'ABR

Mars 2018

Ce sujet est à réaliser en binôme. **3 séances de TP d'1h20** chacune y seront consacrées. La programmation se fera dans le langage de votre choix.

Vous rendrez **les sources de votre projet**, sous la forme d'une archive NOM1-NOM2.zip (où NOM1 et NOM2 sont les noms de famille des étudiants formant le binôme). La décompression de cette archive doit produire un répertoire NOM1-NOM2 contenant tous les éléments de votre travail (compte-rendu, sources, exécutable, jeux d'essai le cas échéant) et un fichier texte contenant les instructions de compilation et d'exécution.

Le compte-rendu doit contenir la description des méthodes utilisées pour les algorithmes, la complexité des fonctions accompagnée de votre argumentaire ainsi que tout autre élément qui vous semble utile à la compréhension de votre travail.

Assurez-vous que vos programmes compilent et fonctionnent sous Linux dans les salles machine du CIE.

Cette archive est à **déposer sous Madoc**. La date limite de restitution est fixée au **Samedi 31 Mars 2018 à 22h00**.

Le non respect des consignes entraîne automatiquement des points en moins. En particulier, les retards seront sanctionnés de la façon suivante : moins d'1 heure de retard : -2 points ; entre 1 et 2 heures de retard : -4 points ; entre 2 et 12 heures de retard : -8 points ; au-delà de 12h de retard : -16 points.

1 Tableau d'ABR (TABR)

Nous considérons dans ce projet une structure de données appelée *tableau d'ABR* (en abrégé : TABR). Il s'agit d'un tableau T dont chaque case i correspond à un intervalle $[T[i].debut; T[i].fin]$, où $T[i].debut$ et $T[i].fin$ sont des entiers. Les informations stockées dans n'importe quelle case i de T sont :

- l'entier $T[i].debut$
- l'entier $T[i].fin$
- un ABR $T[i].arbre$, dans lequel les noeuds contiennent des entiers, tous ces entiers se situant dans l'intervalle $[T[i].debut; T[i].fin]$

De plus, les intervalles ont les propriétés suivantes :

- (a) ils sont bien définis : pour tout i , $T[i].debut \leq T[i].fin$;
- (b) ils sont tous disjoints : deux intervalles différents ne peuvent pas se chevaucher (ils n'ont donc aucun élément en commun) ;
- (c) ils sont ordonnés par ordre croissant : pour tout i (sauf le dernier), $T[i].fin < T[i+1].debut$.

Voir une illustration de la structure TABR à la Figure 1.

2 Travail demandé

- Proposer un programme convivial qui, par l'intermédiaire d'un menu, permet à l'utilisateur de choisir de réaliser une ou plusieurs (à la suite) des fonctions détaillées ci-dessous (Sections 2.1 et 2.2).
- Pour chacune de ces fonctions, indiquer sa complexité, en détaillant vos arguments dans le rapport. On attend ici une complexité sous la forme d'un "grand O" ou d'un "grand Theta", en étant bien entendu le plus précis possible.

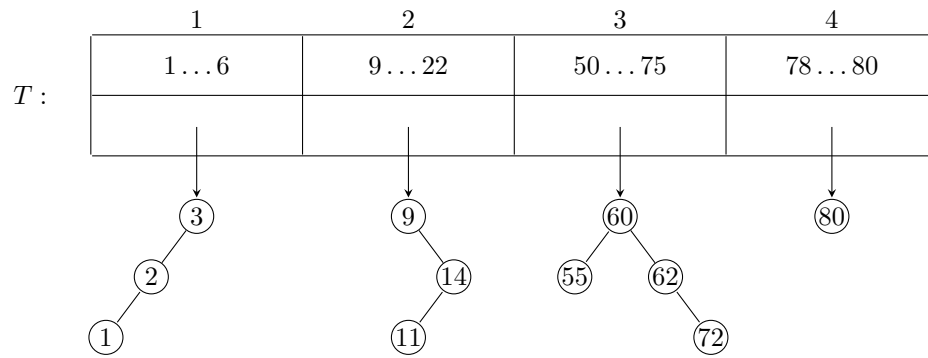


FIGURE 1 – Exemple de TABR

2.1 Génération, sauvegarde et affichage des TABRs

1. **(Fichier vers TABR)** Écrire une fonction permettant de créer un TABR T à partir d'un fichier donné. Chaque ligne i du fichier correspond à la case i de T , et le contenu d'une ligne i est une chaîne

$$a : b; x_1 : x_2 : \dots : x_k$$

telle que :

- $a = T[i].debut$ et $b = T[i].fin$
- l'ABR contenu dans $T[i]$ est le résultat des insertions (dans cet ordre, et partant d'un ABR vide) des valeurs $x_1, x_2, x_3, \dots, x_k$

Par exemple, le contenu du fichier correspondant au TABR de la Figure 1 est :

```
1 : 6 ; 3 : 2 : 1
9 : 22 ; 9 : 14 : 11
50 : 75 ; 60 : 55 : 62 : 72
78 : 80 ; 80
```

2. **(TABR vers fichier)** Écrire une fonction qui effectue la sauvegarde d'un TABR dans un fichier, dont le nom et l'emplacement sont fournis par l'utilisateur. A la suite des valeurs $T[i].debut$ et $T[i].fin$ pour un i donné, on sauvegardera les valeurs des noeuds de l'ABR selon l'ordre obtenu par un parcours préfixe sur cet arbre.
3. **(Affichage sur l'écran)** Écrire une fonction permettant d'afficher un TABR sur l'écran. Par "afficher un TABR" T , on entend reproduire à l'écran ce que contiendrait le fichier correspondant à T .
4. **(Vérification)** Écrire une fonction qui vérifie si un TABR T construit à partir d'un fichier donné en argument est bien rempli, c'est-à-dire que :
 - les intervalles $[T[i].debut; T[i].fin]$ vérifient les propriétés (a), (b) et (c) décrites en Section 1, et
 - $T[i].arbre$ est un ABR dont les éléments sont des entiers contenus dans l'intervalle $[T[i].debut; T[i].fin]$.

2.2 Manipulation des TABRs

5. **(Insertion d'un entier)** Écrire une fonction qui réalise l'insertion d'un entier x dans un TABR T . Cela suppose de trouver l'indice i tel que $[T[i].debut; T[i].fin]$ contient x , puis d'insérer x dans $T[i].arbre$. Si aucun intervalle $[T[i].debut; T[i].fin]$ ne contient x , il ne sera pas inséré et un message l'indiquant sera affiché à l'écran.
6. **(Suppression d'un entier)** Écrire une fonction de suppression d'un entier x d'un TABR. Cela suppose de trouver l'arbre $T[i].arbre$ dans lequel x se trouve (si x est effectivement dans le TABR) et de supprimer x de ce $T[i].arbre$. Si x n'existe pas dans le TABR, on indiquera par un affichage à l'écran dans quel cas on se trouve parmi les deux suivants :
 - a. aucun intervalle ne contient x , ou
 - b. un des intervalles contient x , mais l'ABR correspondant à cet intervalle ne contient pas x .

7. (**Fusion de deux cases du TABR**) Écrire une fonction qui, étant donné un indice $i < n$, transforme un TABR donné T en un TABR T' fusionné comme suit :
- on fusionne les intervalles $[T[i].debut; T[i].fin]$ et $[T[i + 1].debut; T[i + 1].fin]$ en un seul intervalle $[T[i].debut; T[i + 1].fin]$,
 - on fusionne les ABR $T[i].arbre$ et $T[i + 1].arbre$ en un seul ABR,
 - on supprime la case $i + 1$ de T (ceci se fait en décalant toutes les cases de T qui se trouvent à droite de la case éliminée),
 - on met à jour les informations dans la case i ($T[i].debut$, $T[i].fin$ et $T[i].arbre$ doivent correspondre au nouvel intervalle et au nouvel arbre).
8. (**TABR vers ABR**) Écrire une fonction qui, étant donné un TABR T , construit un ABR contenant l'ensemble des éléments des ABR contenus dans T (et aucun élément supplémentaire).