

Mini-projet de TP

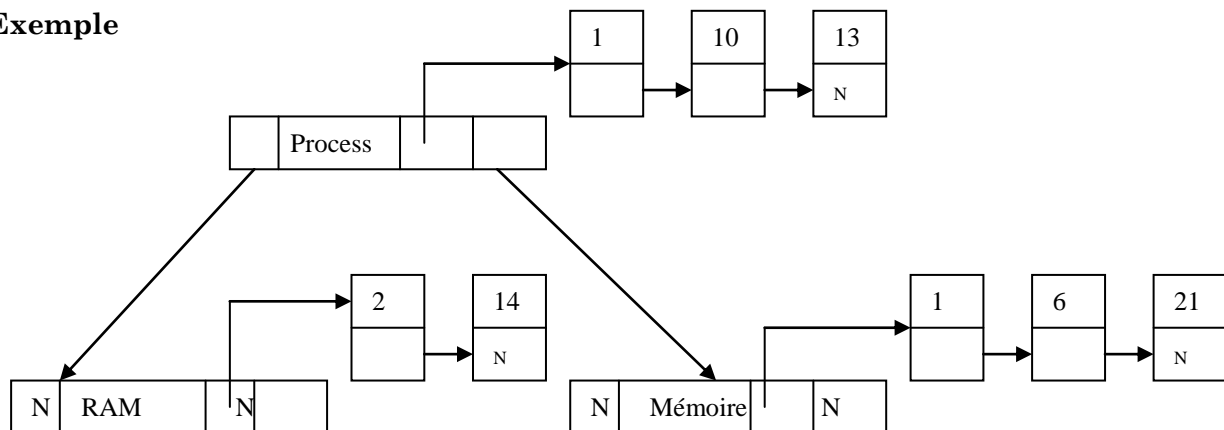
Manipulation de structures dynamiques en C

Objectifs du TP

- Manipulation de structures dynamiques en C (arbres binaires, piles et files)
- Implémentation d'algorithmes de parcours d'arbre binaire et des opérations de manipulation d'arbres binaires, de piles et de files
- Ecriture et appel de fonctions
- Utilisation d'un menu à choix multiples

On voudrait représenter l'index d'un livre par un arbre binaire quelconque (non ordonné) de la manière suivante : Un élément de l'arbre contient un mot qui apparaît dans le livre, ainsi que la liste des numéros de pages où il apparaît (dans l'exemple, le mot « RAM » apparaît dans les pages 2 et 14).

Exemple



Travail demandé

Il s'agit d'implémenter les fonctions répondant aux requêtes suivantes :

Partie I : Opérations sur un arbre binaire non ordonné

1. Construire la structure en introduisant des données « Exemple d'une arborescence avec une dizaine de noeuds » par programme (pour éviter la saisie fastidieuse au clavier le jour du test). Pour chaque mot, on donnera la liste des numéros de pages où le mot apparaît.
Ecrire une fonction **Const-Tree** avec des données fixes et l'appeler dans le programme
2. Construire la structure d'un arbre par saisie de données au clavier.
3. Afficher le contenu de l'arbre (chaque mot avec les numéros de pages associées) en utilisant un parcours en largeur de l'arbre (racine, ensuite les fils de la racine, etc....)
4. Afficher la structure de l'arbre (En précisant pour chaque nœud le fils gauche et le fils droit). Pour chaque mot, on affichera le mot ainsi que la liste des numéros de page entre deux crochets [...] et séparés par des virgules. Par exemple : RAM → [2, 14]

5. Rechercher un mot dans l'arbre (afficher le mot avec la liste des numéros de page ainsi que le niveau de l'arbre où se trouve le mot, le niveau 0 étant la racine)
6. Ajouter un numéro de page pour un mot qui existe déjà dans l'arbre (par exemple ajouter la page 15 pour le mot "Mémoire" dans l'arbre, les numéros de pages doivent rester triés)
7. Supprimer un numéro de page *p* pour un mot *M* donné (s'il existe).

Partie II Opérations sur un arbre binaire ordonné (ou arbre binaire de recherche)

Dans cette partie, on reprend la structure de la partie I. On désire construire un arbre binaire ordonné à partir d'un arbre binaire non ordonné et effectuer des opérations sur l'arbre binaire ordonné. Dans un arbre binaire ordonné, le fils gauche d'un nœud **a** contient un mot plus petit (selon l'ordre alphabétique) et le fils droit contient un mot plus grand que celui contenu dans **a**.

Par exemple le mot « système » doit être à droite (fils-droit) du mot « ordinateur »

le mot « mémoire » doit être à gauche (fils-gauche) du mot « ordinateur »

1. A partir d'un arbre d'index non ordonné déjà construit, créer un arbre binaire ordonné et équilibré (on pourra utiliser une structure intermédiaire).
2. Afficher le contenu de l'index en respectant l'ordre croissant des mots et des pages pour chaque mot.
3. Rechercher un mot donné dans l'arbre et retourner son niveau (s'il existe).
4. Ajouter un mot donné dans l'arbre avec un numéro de page. Si le mot existe déjà, on ajoute uniquement le numéro de page
5. Supprimer un mot de l'index avec tous les numéros de pages associées au mot.

Partie III Chemin dans un arbre binaire ordonné

Dans cette partie, on désire utiliser une pile pour construire tous les chemins de l'arbre. Un chemin est une branche contenant tous les nœuds allant de la racine jusqu'à un nœud feuille. On peut représenter un chemin par une liste chaînée (ou une file). Dans ce cas, ça sera une suite de mots contenus dans les nœuds de l'arbre.

1. Construire les différentes listes comportant les chemins de l'arbre
2. Afficher tous les chemins en commençant par le plus à gauche.

Indications

- L'interface principale du programme doit se présenter sous forme d'un menu à choix multiples pour répondre à l'une ou l'autre des requêtes de l'utilisateur.
- Le programme doit être écrit sous forme de fonctions.
- Les affichages des résultats à l'écran doivent être lisibles et bien faits.
- Les messages affichés à l'utilisateur pour l'introduction de données doivent être clairs.

Remarques importantes

Le code source doit être :

- **Bien commenté**, une ligne de commentaire est obligatoire avant chaque fonction, indiquant ce que fait la fonction plus d'autres commentaires jugés nécessaires.
- Les noms des étudiants doivent être indiqués au début du programme.

Rapport à remettre

Rédiger un rapport comportant une description des différentes structures de données utilisées et les différents algorithmes implémentés pour chacune des parties.

Indications - Pour la mise en forme :

Page de garde :

Une page de garde est nécessaire et doit comporter :

- En haut, au centre: le nom de l'université, la faculté et le département (police Times New Roman, 11) + le Logo de l'USTHB.
- En dessous (avec espacement) La mention « Rapport de Mini projet de TP Algorithmique et Structures de données – Programmation C », (Times New Roman, 14)
- Au milieu de la page, le titre du projet « Manipulation de structures dynamiques... », Times New Roman 16, gras
- En dessous à gauche : Réalisé par : Noms et prénoms des étudiants, Times New Roman 12
- A droite : Nom et prénom des enseignants : Responsable du module et enseignant de TP
- En bas de la page, au centre, Année : 2022/2023, Times New Roman, 10.

Les autres pages

Il faut une introduction, objectifs du TP,...

Il faut une conclusion à la fin

Il faut des titres de sections pour séparer les différentes parties

Marges : 2,5cm de tous les côtés (haut, bas, à gauche et à droite)

Police de caractères (11 ou 12 Times New Roman)

Interligne (multiple de 1,3)

Titres de sections : Times New Roman (13 ; gras)

NB

- | |
|---|
| <ul style="list-style-type: none">○ Le travail doit être fait <u>par binôme</u> et envoyé à l'adresse works1819@gmail.com le 7 janvier 2023.○ Le code source devra être dans un fichier WordPad○ Le compte rendu devra être dans un fichier pdf○ Les noms des étudiants doivent figurer dans les deux noms de fichiers○ L'objet du mail : ProjetTP-Algo-Noms des étudiants○ Aucun retard ne sera toléré○ Les cas de copiage seront sanctionnés par un zéro. |
|---|