

Séance 6: tables de hachage

Les objectifs de cette séance, et du travail individuel à fournir d'ici la séance suivante, sont :

- 4-Compiler des projets composés de plusieurs fichiers .c et .h
- 6-Être capable d'implémenter concrètement un ADT donné, de plusieurs manières
- 17-Utiliser l'ADT "tableau associatif" pour résoudre des problèmes concrets en C
- 18-Comprendre le fonctionnement d'une table de hachage
- 19-Implémenter un tableau associatif au moyen d'une table de hachage
- 20-Manipuler des fichiers texte en C

1. Théorie sur la table de hachage

Une manière d'implémenter un tableau associatif consiste à employer une table de hachage, qui est un tableau associé à une fonction qui permet de faire correspondre chaque clé à un entier positif correspondant à une case du tableau (fonction de hachage).

1.1. Insérer un élément dans une table de hachage

Imagine une table de hachage constituée d'un tableau de 10 cases, dont les clés sont des chaînes de caractères, dont les valeurs sont des entiers et dont la fonction de hachage est la suivante :

$\text{hachage}(\text{char}^* \rightarrow \text{int})(x) = (\text{somme de la valeur ASCII de chaque caractère de } x) \% 10$

97	61	a	111	6F	o
98	62	b	112	70	p
99	63	c	113	71	q
100	64	d	114	72	r
101	65	e	115	73	s
102	66	f	116	74	t
103	67	g	117	75	u
104	68	h	118	76	v
105	69	i	119	77	w
106	6A	j	120	78	x
107	6B	k	121	79	y
108	6C	l	122	7A	z
109	6D	m			
110	6E	n			

Dessine et explique l'ajout successif des éléments (clé, valeur) suivants : (luc, 29), (ced, 36), (tom, 35), (io, 40), (seb, 39). Quel problème se pose ? Comment pourrais-tu le résoudre ? Propose deux solutions : une basée sur le "chainage", et une basée sur "l'adressage ouvert". Puis explique le concept de "taux de remplissage".

1.2. Récupérer un élément dans la table de hachage

Explique comment se passe la récupération d'un élément dans la table de hachage, pour les deux solutions que tu as proposées à la question précédente.

1.3. L'importance de la fonction de hachage

Une bonne fonction de hachage est essentielle pour garantir que notre table de hachage soit remplie de la façon la plus efficace possible. Déterminez ce qu'est une fonction de hachage uniforme. Donnez des exemples de deux fonctions de hachage, dont l'une serait moins uniforme que l'autre. Expliquez pourquoi.

1.4. Complexité des opérations

Évaluez la complexité des quatre opérations d'un tableau associatif implémentées au moyen d'une table de hachage, en supposant que la fonction de hachage soit très efficace (de l'ordre de $O(1)$) et qu'elle soit très uniforme (et donc que les éléments sont très bien répartis dans le tableau).

2. Implémentation d'un tableau associatif

2.1. Table de hachage

Définissez la structure concrète table de hachage, que vous emploieriez pour implémenter l'ADT tableau associatif. En particulier, définissez une fonction de hachage, une fonction pour créer une nouvelle table de hachage, pour ajouter des éléments, supprimer des éléments, récupérer des éléments, une fonction pour agrandir la table si elle est trop remplie (par exemple lorsque le taux de remplissage est de $2/3$) etc.

Déterminez une fonction de hachage efficace et suffisamment uniforme. Justifiez.

2.2. Tableau associatif

Implémentez maintenant votre tableau associatif au moyen de la structure concrète table de hachage que vous avez définie précédemment.

3.Lecture de fichiers

3.1.Ouvrir et lire un fichier en C

Depuis un programme C, vous pouvez ouvrir des fichier ".txt" au moyen de la fonction "fopen" qui renvoie une valeur de type FILE. La fonction "gets" permet la lecture du fichier ligne par ligne. La fonction "fclose" permet de fermer le fichier lorsque tu as terminé la lecture.

Essaie d'ouvrir le fichier nana.txt et d'en afficher les deux premières lignes.

Essaie de stocker en mémoire la totalité du texte.

3.2.Lister les mots d'un texte

Écrivez un programme qui vous permet d'obtenir la liste des mots présents dans un fichier de texte, avec à chaque fois son nombre d'occurrences de ce mot. Par exemple, une fois le fichier analysé, l'utilisateur peut entrer un mot et le programme lui indique combien d'occurrence de ce mot il y a dans le texte. Petits conseils : détermine la liste des séparateurs de mots (espaces, ponctuation) et pour chaque mot, transforme tout en minuscules. Et utilise un tableau associatif implémenté au moyen d'une table de hachage !