

Laboratoire #9

Les tables de dispersion

L'objectif de ce laboratoire est de vous familiariser avec le concept des tables de hachages. Après ce travail, vous devriez être en mesure de comprendre les éléments suivants :

- Fonctions de dispersion
- Implémentation d'une table de hachage
- Techniques de résolution de collisions

Documentations : voir la section Documentation/**Normes de programmation** sur le site Web du cours. Vous y trouverez la description des normes de programmation en vigueur dans notre cours.

Travail à faire

Les tables de hachage permettent d'obtenir un temps de recherche moyen d'ordre $O(1)$, à condition que la fonction de dispersement soit bien choisie. Dans le présent laboratoire, on vous demande d'implémenter une table de hachage par **adressage ouvert et sondage quadratique** pour la gestion des collisions et d'étudier son efficacité.

1. **Implantation.** Considérez la définition de la classe HashTable fournie dans l'archive HashTable.zip, vous devez compléter le fichier *HashTable.hpp* fourni également par les implémentations manquantes. Notez que vous pouvez ajouter autant de méthodes privées que vous voulez. Nous vous fournissons un *main()* dans le fichier *TestHashTable.cpp* afin de tester votre travail et que vous pouvez évidemment enrichir.

Notez bien. Le *main()* fourni est minimaliste, vous devez enrichir les tests par de multiples insertions et suppressions.

2. **Tests.** Vous devez modifier la classe HashTable afin de vous permettre d'obtenir le nombre d'opérations nécessaires lors de la recherche d'un élément. Tester l'efficacité de cette table. Pour ce faire, suivez les étapes suivantes :
 - En premier lieu, insérez dans une table de 1013 cases 500 éléments avec une valeur de clé, entière cette fois, aléatoire de situant dans l'intervalle $[10000, 19999]$. Considérer la fonction de hachage $H_1()$ ci-après.
 - Recherchez 1000 éléments aléatoires et calculez la moyenne d'opérations nécessaires pour chaque recherche.
 - En deuxième lieu, on vous demande de modifier la classe HashTable afin de supporter la technique du double hachage telle que présentée dans le cours. Afin d'assurer une meilleure adaptabilité, les fonctions de dispersement utilisées par la classe doivent être des foncteurs comme c'est le cas présentement. Tester à nouveau l'efficacité de la table en suivant la même technique. Utiliser les fonctions de hachage suivantes :

$$Position = (H_1(x) + iH_2(x))\%N, \text{ où}$$

$$H_1(x) = x\%N, \text{ et } H_2(x) = (919 - (x\%919))\%N$$

Vous êtes tenu de faire la gestion des exceptions dans les méthodes que vous avez à implanter et de les documenter dans le format de Doxygen.

Bien livrable :

Vous devez rendre un fichier .zip comportant uniquement les fichiers suivants :

L'ensemble des fichiers fournis et demandés complétés selon les normes de programmation en vigueur dans le cours, avec les commentaires d'interface en format Doxygen.

- Les fichiers complétés avec les commentaires d'implantation en format Doxygen.
- Le fichier doxyfile permettant la génération de la documentation.
- prévoyez deux versions de la template HashTable :
 - HashTable pour l'implantation avec hachage par adressage ouvert et sondage quadratique
 - et HashTableD pour l'implantation supportant la technique du double hachage

Vous devez remettre votre travail en utilisant votre guichet étudiant de Pixel. (<https://pixel.fsg.ulaval.ca/>), rubrique remise de TP. Ce travail est intitulé Laboratoire 9. **Aucune remise par courriel n'est acceptée.**

Date de remise

Ce travail est à remettre au plus tard le **lundi 25 novembre 2013 9h00** (par intranet uniquement, voir ci-dessus).

Pour tout retard non motivé (voir plan de cours; motifs acceptables pour s'absenter à un examen), la note 0 sera attribuée.