

Projet 2 : Chariots (3 séances, à rendre)



Ce projet continue à faire implémenter une simulation de super-marché dans lequel un client remplit un chariot (caddie™) avec des produits puis passe en caisse pour payer. Cette fois-ci, on s'intéresse à modéliser le chariot : le client peut ajouter dans son chariot un produit qu'il prend dans un rayon du magasin mais le range toujours (c'est un modèle:-) pour mettre les plus lourds au fond du caddie (et donc les plus légers au dessus).

Par conséquent, il reprendra en caisse les produits du chariot en commençant par les plus légers.

Un premier travail d'analyse a été effectué et une implémentation utilisant une file de priorité est envisagée.

Votre travail consiste à définir des structures de File de priorité et à les comparer en termes de coût spatial et temporel. Vous devez aussi compléter la classe chariot.

Spécification de file de priorité :

```
class FP{
    FP();           /* Constructeur, crée une FP vide */
    ~FP();          /* Destructeur, libère la mémoire */
    void enfiler(const T & elt); /* Ajout d'un élément */
    bool estVide() const;      /* Test de l'absence d'éléments */
    const T & premier() const; /* accès au premier élément de la file */
    void defiler();           /* Suppression du premier élément */
    unsigned int taille() const; /* Taille de la file */
};
```

Fichiers fournis :

Vous trouverez sur Madoc une archive contenant :

- une classe Produit à **ne pas modifier** (produit.hpp + produit.cpp) implémentant les produits vendus dans le magasin ;
- une classe Mappe à **ne pas modifier** (mappe.hpp + mappe.hpp) implémentant une table associative ;
- une classe Magasin (magasin.hpp + magasin.cpp — à **ne pas modifier** —) prenant en paramètre la classe implémentant la table associative ;
- une classe Chariot (chariot.hpp — à **ne pas modifier** — + chariot.hpp) prenant en paramètre la classe implémentant la file de priorité ;
- un squelette de programme de test de cette classe (chariot_test.cpp) que vous complétez pour vérifier que le chariot se comporte de la même façon avec les deux implémentations que vous allez écrire et les comparer ;
- un makefile pour compiler (utiliser make all dans le dossier).

Travail à effectuer :

Commencez par prendre connaissance du contexte : lisez intégralement le sujet (;-) puis étudiez le code fourni. Profitez-en pour noter dans un document de travail les questions que vous vous posez, les remarques qui vous viennent à l'esprit, les idées d'amélioration, tout cela (une partie tout au moins) nourrira votre rapport. Ce document présentera aussi les choix effectués (constantes, comportement des méthodes non précisé dans l'énoncé, ...) et les limites d'utilisation. Il est aussi, si ce n'est plus important, que le code.

Adaptez le code de votre classe FPtas (TP précédent) pour qu'elle respecte la spécification ci-dessus.

Complétez le code de la classe Chariot : il manque dans chariot.hpp l'implémentation des fonctions.

Complétez le programme de test de chariot chariot_test.cpp pour les vérifier avec votre structure (il suffit de préciser la classe passée en paramètre du constructeur).

Définissez une seconde implémentation de File de Priorité (différente bien sûr).

Comparez les temps d'exécution de vos structures, sur des exemples pertinents ; faites des graphiques.

Documents à rendre :

Une archive contenant le code C++ utilisé et le rapport.

- Pour le code, vous pouvez simplement créer une archive du dossier projet2 complet (supprimer les exécutables pour gagner de la place). Il doit y figurer le code de vos classes (Files de priorités) et de celle complétée (chariot.hpp) ainsi que des programme de test (chariot_test.cpp).
- Le rapport, impérativement au format pdf, devra comporter un comparatif des deux implémentations en terme de complexité temporelle (l'ordre de grandeur du coût de chaque méthode doit être renseigné) et si possible d'encombrement mémoire. D'autres implémentations (suffisamment différentes) peuvent y être présentées (la réalisation n'en est pas demandée) et comparées.

Le travail est à déposer sur MADOC au plus tard le 31 mars 2015 à 23h55 sous forme d'une archive zip contenant le rapport (pdf) et le code (archive du dossier complet) .

Le code doit être clair et concis, commenté, avec en particulier les pré- et post-conditions. Le rapport doit être bien écrit (français correct, attention à l'orthographe !) et agréable à lire, il doit comporter introduction et conclusion et des annexes pour les détails techniques (graphiques, code, etc.)

