

M1 ALMA
Université de Nantes
2011-2012

Rapport de Projet : Programmation générative

MARGUERITE Alain
RINCÉ Romain

Université de Nantes
2 rue de la Houssinière, BP92208, F-44322 Nantes cedex 03, FRANCE

Table des matières

Table des matières	1
1 Concrète	2
1.1 Developpement	2
Table des figures	4

1 Choix de la structure concrète

Concrète

Nous avons recherché la structure concrète la plus performante dans l'ensemble des opérations. Ne pouvant savoir quelle serait la nature de l'utilisation de cette classe, la structure ayant les une moyenne des performance la plus forte sur l'ensemble des candidates, semblait un bon compromis. Plusieurs solutions se présentaient pour une implémentation de tas binaire :

Tableau des pères Non approprié (dédié aux arborescence non-ordonnées).

Une liste d'adjacence Un noeud étant un élément (étiquette, père, liste des fils) de la liste. Ses performances sont dégradées lors d'opération comme la suppression ($O(n)$ pour un tas binaire) ou l'accès à un noeud à partir de son étiquette ($O(n)$). Dans le cas d'une utilisation où les opérations principales aurait été l'ajout d'un fils par exemple, cette solution aurait pu être retenue.

Liste de fratrie Un noeud étant un élément (étiquette, père, liste des frères, premier fils) de la liste.

Tableau modulaire definition manquante Structure présentant les meilleurs complexités pour l'ensemble des opérations. Cette solution répond le mieux aux contraintes que nous avons exposés précédement. C'est donc elle que nous avons retenue.

1.1 Developpement du tas binaire

Une fois la selection de la structure faite. Nous avons au cours des semaines adopté le plan de developpement suivant :

1. Définition des prototypes des méthodes
2. Implémentation des méthodes
3. Rajout de l'itérateur
4. Rajout de l'allocateur et du comparator
5. Debug

Cette démarche nous a permis d'avancer rapidement au cours des première semaine. Les cours d'algorithmiques du 1^{er} semestre encore en tête. Nous avons rapidement déployées les méthodes principales du tas binaire (Extraire, Tasser, Insérer...), sans trop nous soucier de la syntaxe C++ (nouvelle pour notre bînome). Nous avons amèrement regretté notre insouciance de ne jamais tester nos méthodes avant la 5^{ème} ou 6^{ème} semaine. Nous avons passé un temps très long à déboguer un nombre de d'erreurs de syntaxe conséquent.

Table des figures