

Algorithme et Structures de Données (ASD)

Plan du Cours

Dr Ibrahima Gaye

Université Alioune DIOP de Bambey (UADB)

9 janvier 2021

- ☞ les Types Abstraits de Données (TAD) ou Aype Agébrique de Données (TAD)
- ☞ les structures séquentielles (Listes, Cas particuliers (Piles et Files))
- ☞ les ensembles
- ☞ les structures arborescentes

les types abstraits (1)

La conception d'un algorithme peut se faire de deux manières :

- ☞ La première et aussi la plus simple est celle qui consiste à coder et à considérer uniquement les types de données créés pour l'occasion et correspondants au langage de programmation utilisé. Leur implémentation se fera via les déclarations de types du langage et l'on utilisera alors les méthodes (procédures et fonctions) qui leur sont propres. Il est alors parfois difficile de transcrire facilement l'algorithme vers un autre langage (du C vers le Java par exemple ou réciproquement).

les types abstraits (2)

- ❏ La deuxième méthode consiste à définir des types de données abstraits. C'est à dire que l'on va définir pour un type de donnée son nom, sa syntaxe d'utilisation (opérations, paramètres, etc.) et ses propriétés tout en étant détachés de toutes les contingences propres à un langage de programmation et/ou à une machine. Si, par exemple, un problème nécessite l'utilisation d'un graphe, nous nous contenterons de penser à la résolution du problème sans nous soucier du comment sera implémenté ce graphe en mémoire, ni comment seront manipulées ses différentes composantes.

les structures séquentielles

Nous présenterons dans ce chapitre les structures séquentielles "classiques", à savoir les listes, les piles et les files. Nous donnerons à chaque fois la définition du type abstrait correspondant accompagnée de quelques exemples explicatifs. Puis nous fournirons les diverses implémentations de ces types ainsi que quelques algorithmes de manipulation. Les algorithmes seront fournis à chaque fois et dans la mesure du possible des deux manières : abstraite (en utilisant les opérations des types abstraits) et concrète.

Contrairement aux structures séquentielles comme les listes où l'ordre des éléments est fondamental, pour les ensembles c'est une notion sans importance. Ce qui importe, c'est qu'un élément soit présent ou non dans la structure. Ce chapitre ne fera pas de rappel sur les définitions et propriétés des ensembles qui sont supposées connues. Cela étant il présentera aussi les familles (multi-ensembles ou ensemble à répétitions) qui acceptent les occurrences multiples d'élément au sein d'un ensemble.

les structures arborescentes

- ☞ Un arbre est constitué d'éléments appelés nœuds organisés de façon hiérarchique. L'élément de base est la racine. En informatique, comme dans la vraie vie, les arbres se rencontrent un peu partout : construction de répertoire, système d'exploitation, compilateur, etc.
- ☞ Du à leur propriété structurelle récursive, les manipulations et les déclarations des arbres se font naturellement sous une forme récursive. Ce qui ne veut pas dire que la forme itérative ne leur soit pas applicable, mais elle est moins intuitive et souvent moins adaptée.
- ☞ Les arbres se rencontrent à la base sous deux formes : les arbres binaires et les arbres généraux.