

# Algorithmique & Programmation

## Liste des thèmes de l'examen oral de juin 2016

### MODALITÉS

- Vous tirez au hasard une question d'Algorithmique.
- Pour la partie Programmation, vous serez interrogés sur l'un des travaux pratiques que vous avez rendus au cours du semestre. Le jury vous indiquera avant la préparation sur quel travail vous serez interrogés. La version exacte du travail que vous avez rendu servira de base à l'examen. Vous n'avez pas besoin d'imprimer vos travaux pratiques ; pendant l'examen, nous nous baserons sur la version électronique rendue.
  - Vous deviez réaliser 18 séries de travaux pratiques, chaque série comptant pour 1 point, sauf la série A07 qui compte pour 2 points. Comme vous ne deviez rendre que 75% de ces travaux, vous pouvez choisir les travaux sur lesquels vous serez interrogés ; les travaux choisis doivent totaliser 12 points (il y a 19 points possibles, 12 représente donc le 63.16% des points possibles).
  - Les étudiants qui ont réalisé plus de 75% des travaux pratiques dans les délais peuvent effectuer leur choix librement comme décrit ci-dessus parmi les travaux qu'ils ont rendu.
  - Les autres doivent obligatoirement présenter l'ensemble des travaux qu'ils ont rendu dans les délais.
- Vous avez 30 minutes pour préparer (sans autre document que la version électronique du travail pratique choisi par le jury) un exposé d'une dizaine de minutes traitant la question d'Algorithmique ainsi que l'interrogation de programmation.
- Durant l'exposé d'Algorithmique, **les membres du jury n'interviennent en principe pas**.
- Vous êtes interrogés pendant une dizaine de minutes sur le travail pratique choisi par le jury.
- Le temps restant (5 à 10 minutes) sera consacré à des questions du jury ; ces questions concerneront certains points évoqués dans votre exposé ainsi que des aspects de la question que votre exposé aurait éventuellement omis.
- Vous disposez du tableau pour noter tout ce que vous jugerez utile à la clarté de votre exposé (plan, définitions, propositions ou exemples importants, éventuellement représentations graphiques ou dessins).
- Dans l'évaluation de votre prestation, il sera tenu compte de la complétude et de la structure de votre exposé, de sa pertinence et de son adéquation à la question tirée ainsi que de la qualité de vos réponses aux questions du jury.

### THÈMES D'ALGORITHMIQUE

**N.B.** : Un thème peut donner lieu à **plusieurs** questions et **une** question peut porter sur **plusieurs** thèmes !

#### ➤ L'ALLOCATION DE MÉMOIRE

- Les divers modes d'allocation (statique et dynamique : généralités).  
L'allocation automatique en détails (*ce qui se passe lors de l'appel d'une procédure*).
- Les divers modes de passage de paramètres et leur disponibilité dans divers langages de programmation (essentiellement la distinction entre la famille C et la famille Pascal).
- La dynamique du **Stack** lors de l'appel de procédure en général.  
La spécificité du cas de la récursivité.

#### ➤ NOTIONS GÉNÉRALES SUR LA RÉCURSIVITÉ

- Principes généraux (critère de récursivité, étapes de l'analyse récursive, les difficultés de la clause de baisse de complexité, liens avec le **Stack**, performance, ...).
- Récursivité et formants de boucle : Exemples et discussion de la "dérécursivisation\*"  
(avantages, inconvénients, facilité de réalisation).  
Liens avec la notion de **récursivité terminale**.

\*Transformation d'un algorithme récursif en algorithme sans appel récursif, conçu exclusivement au moyen de formants de boucle.

## ➤ LES PROBLÈMES NATURELLEMENT RÉCURSIFS

**Attention** : Il s'agit de connaître l'**énoncé initial** du problème et de savoir en **déduire explicitement** la résolution récursive ; prendre cette solution comme **donnée** sera **absolument insuffisant** !

Par exemple, pour Fibo, la donnée est réduite aux propriétés des lapins-de-Fibonacci (immortels, stériles pendant les deux 1<sup>ères</sup> périodes de vie, produisant ensuite un couple semblable à chaque période) ; votre exposé doit partir exclusivement de cette donnée pour aboutir démonstrativement à " $F(n+2) = F(n+1) + F(n)$ " et en aucun cas partir de cette formule comme si elle était donnée au départ !

- Les Tours de Hanoï.
- Les permutations (→ Factorielle).
- Les coloriages de boules.
- Le dénombrement des arbres arithmétiques de n feuilles (→ Fonction de Catalan).
- Les tris dichotomiques en général et **QuickSort** en particulier (avec ses **nombreuses** caractéristiques spéciales ...).
- Les lapins de Fibonacci (→ suite de Fibonacci).
- Les combinaisons (→ Coefficients binomiaux).

## ➤ PILE ET QUEUE EN GÉNÉRAL

- En tant que **module** de gestion chronologique de l'information (comme *Object*, avec les méthodes et les attributs caractéristiques et essentiels : Implantation **s** et pratique).
- En tant que **mode** de gestion chronologique de l'information (exemples de gestion, en général et en particulier dans le cadre informatique).

## ➤ LES EXPRESSIONS ARITHMÉTIQUES

- Les trois notations : Définitions, propriétés particulières, conditions de validité syntaxique (lien avec la validation d'expressions plus générales, en particulier celles d'un langage de programmation), illustration et traductions grâce à l'arbre arithmétique correspondant.
- La notation polonaise inverse en grands détails, son statut particulier pour les calculateurs automatiques et son évaluation au moyen d'une **pile arithmétique**.

## ➤ LES STRUCTURES DÉFINIES RÉCURSIVEMENT

### - **La liste**

Version abstraite : Définition et explication du rôle de la récursivité.

Implantation minimale (celle où nul l est une liste vide valide) : Réalisation, illustration, discussion, connaissance parfaite (réalisation et illustration) des méthodes fondamentales (ajout, destruction, dénombrement, déroulements).

Problématique de l'emploi de cette forme minimale : Justification de la création de la *wrapping class* associée avec implantation par délégation ; forme et utilisation de cette *wrapping class*.

Emploi pour implanter une Pile ou une Queue : Discussion, illustrations et réalisations. Définition, illustration et explication des avantages de la **tête de liste**.

### - **L'arbre**

Version abstraite : Définitions, terminologie (arité, racine, fils, ordonné), illustrations distinctions et explication du rôle de la récursivité.

### - **L'arbre binaire**

Statut particulier, **théorème du binaire équivalent**.

Implantation minimale (celle où nul l est un arbre vide valide) : Réalisation, illustration, discussion, connaissance de quelques méthodes de construction et maîtrise des déroulements (lien et distinction avec les arbres arithmétiques) : Liens entre les définitions récursives des déroulements et celles qui utilisent **l'enveloppe** de l'arbre (le petit chemin vert qui fait tout le tour de l'arbre).

### - **L'arbre de recherche**

Définition, propriétés, finalité, mode d'emploi et illustration.

## **THÈMES DE PROGRAMMATION**

(Ces thèmes ont tous été traités dans le cours ainsi que dans un ou plusieurs des travaux pratiques qui vous ont été proposés; ils sont donc susceptibles d'être abordés dans le cadre de l'interrogation sur le travail pratique choisi par le jury)

- |   |   |
|---|---|
| 1. Architecture <i>JDBC</i>                             | 9. Méthodes génériques                                  |
| 2. « Mapping » objet-relationnel                        | 10. Autres nouveautés du langage dès Java 1.5           |
| 3. Accès à la base de données                           | 11. Création de nouveaux composants graphiques          |
| 4. Concept de <i>package</i>                            | 12. Composants dérivés d'un composant existant          |
| 5. Architecture d'une application                       | 13. Composants existants assemblés dans un <b>Panel</b> |
| 6. Mise en œuvre de l'héritage                          | 14. Protocole <i>JavaBeans</i>                          |
| 7. Classes paramétrées                                  | 15. Intégration des composants à l'IDE                  |
| 8. Compatibilité en assignation des classes paramétrées | 16. Événements et mise en œuvre dans un composant       |