Programme de Colle N°6 (Semaine du 20/11/2023)

19 novembre 2023

Questions de Cours

1. Donner la définition d'une fonction g majorée par f à une constante près, minorée par f à une constante près, de même ordre de grandeur que f et équivalente à f, avec f et g deux fonctions de $\mathbb{N}$ dans $\mathbb{N}$.
2. Ecrire l'algorithme d'exponentiation rapide en utilisant la récursivité.
3. Ecrire l'algorithme d'exponentiation naïve en utilisant la récursivité. Montrer que cet algorithme termine et est correct.
4. Définir en OCaml des fonctions `longueur`, `somme` et `max` qui prennent en entrée une liste et renvoient respectivement la longueur, la somme des éléments et le maximum de la liste.
5. Définir en OCaml une fonction `tri_fusion` qui prend en entrée une liste et renvoie la liste triée à l'aide de l'algorithme de tri fusion. On pourra utiliser des fonctions `division` pour diviser une liste en deux sous-listes de même taille (à un élément près) contenant les éléments de la liste initiale, et une fonction `fusion` pour fusionner deux listes triées.
6. Définir en OCaml une fonction `tri_insertion` qui prend en entrée une liste et renvoie la liste triée à l'aide de l'algorithme de tri par insertion. On pourra utiliser une fonction `insertion` pour insérer un élément dans une liste triée.

Cours

Chapitre 3

— Terminaison d'un algorithme. Variant de boucle, preuves de terminaison.

— Correction partielle ou totale d'un algorithme. Invariant de boucle, preuve de correction.

— Complexité temporelle : définition. Nombre d'opérations élémentaires pour une entrée donnée. Taille d'une entrée, complexité dans le pire cas et en moyenne pour une entrée de taille n.

— Notations de Landau, complexité asymptotique : fonction majorée par/minorée par/de même ordre de grandeur que/équivalente à f.

— Complexités usuelles, règles de calcul sur les complexités. Sommes usuelles. Analyse de complexité.

Chapitre 4

- Définition d'une fonction récursive. Syntaxe en C. Définition récursive des fonctions factorielle et puissance (exponentiation version naïve et rapide)
- Cas terminal d'une fonction récursive. Arbre d'appels.
- Comparaison des méthodes récursives et itératives. Exemple d'application de la récursivité : tours de Hanoi.
- Syntaxe du langage OCaml :
  - Déclaration de variables, types.
  - Conditions, déclarations locales.
  - Déclaration de Fonctions. Filtrage par Motifs.
  - Traits Impératifs.
  - Types Construits.
- Types définis récursivement.
- Récursion Mutuelle.
- Pile d'Appels.
- Terminaison, Correction et Complexité des fonctions récursives (Exemples élémentaires).

TP4

- Mutabilité en OCaml :
  - Enregistrements modifiables
  - Références
  - Tableaux
  - Représentation des Matrices par les tableaux