

Semestre 1 & retour sur trace

Programme de colle

Semaine 19 (03 février 2025 – 08 février 2024)

Programme

Semestre 1

- Preuve de terminaison. Preuve de correction.
- Preuve de complexité pire des cas. L'analyse des fonctions récursives se limite à des cas simples.
- Preuve de complexité amortie. Ex : tableaux dynamiques, compteur binaire.
- Tableaux dynamiques, listes chaînées.
- Piles, implémentées comme des listes simplement chaînées ou des tableaux (éventuellement dynamiques).
- Files, implémentées comme des listes simplement chaînées ou des tableaux circulaires.

Retour sur trace

- Exploration exhaustive : concept, limites de l'approche.
- Exploration exhaustive par une fonction récursive, en construisant une solution choix après choix.
- Fonction de rejet et retour sur trace. *L'importance de l'ordre dans lequel on fait les choix a été illustré sur des exemples, mais n'est pas un attendu.*
- Limites de l'analyse de complexité pire des cas pour un retour sur trace.
- Exemples traités : n dames, subset sum (avec des entiers positifs), tour ouvert du cavalier d'Euler, mot ternaire sans carré.

Induction

L'objectif n'est pas de faire de la théorie des ordres, mais de reconnaître les contextes où l'on peut faire un raisonnement par induction et de savoir en rédiger un.

- Relation d'équivalence.
- Relation d'ordre. Prédecesseur, prédecesseur strict, prédecesseur immédiat. Ordre partiel, ordre total.
- Relation d'ordre bien fondée.
- Thm : on peut faire des raisonnements par induction si et seulement si l'ordre est bien fondé.
- Ex : nombre d'objets dans un mobile de Calder (un arbre binaire strict à f feuilles a $f - 1$ noeuds internes).

Les arbres n'ont pas encore été vus.

Et plus

Tout ce qui a été fait depuis le début de l'année est au programme. De plus, certains exercices peuvent tout à fait s'écarter du programme pour tester votre réactivité face à l'inconnu.

Questions de cours

Toute colle commencera par une de ces questions de cours, notée sur 10/20.

Une tolérance vis-à-vis de la syntaxe pourra être appliquée sur les questions d'implémentations si l'élève n'est pas sur machine.

- Exponentiation rapide : code (C ou OCaml), analyse de la complexité de l'exponentiation rapide. On ne comptera que les multiplications comme opérations, dont on calculera le nombre exact. On se limitera au cas où la puissance est une puissance de 2.
- Compteur binaire : analyse de la complexité d'un INCR au sein d'une suite arbitrairement longue de INCR sur un même compteur binaire, par la méthode du choix de l'élève (état initial : compteur à 0). On ne traite pas le cas d'overflow.
- Files : schéma explicatif de l'implémentation par tableau circulaire d'une file. Proposer un type `struct` correspondant en C, et le code de `enfile` ou `defile`. *La difficulté liée à distinguer la file pleine de la file vide peut-être ignorée.*
- Retour sur trace : proposer un algorithme de retour sur trace pour SubsetSum en OCaml ou C, et un critère de rejet (quand tous les entiers sont positifs). On ne cherchera pas à renvoyer une solution valide, simplement à en déterminer l'existence. *Le problème SubsetSum peut, au besoin, être rappelé pour le colleur.*