Colle Bonus : MP2I

MP2I – Lycée Louis Thuillier

Introduction

Cette planche de colle explore les fondamentaux de la programmation et de la complexité :

- La discipline de programmation : spécification, préconditions et postconditions,
- La programmation défensive et les assertions,
- La pile d'appels, la preuve et l'analyse de complexité des algorithmes récursifs,
- L'introduction à la programmation en OCAML.

Les exercices sont conçus pour tester votre compréhension, votre rigueur et votre capacité d'analyse.

Exercices

1.1 Exercices de cours

Exercice

Exercice de cours 1.

- Après avoir définie les termes en **gras**, répondre à la question suivante :
- Dire quel(s) algorithme(s) sont **totalemment correctes** et lesquels sont **partiellement correctes**.

```
1.
1 let rec contient l x = match l with
2 | [a] when a=x-> true
3 | e::t -> contient t x;;
```

```
2.
1 let rec contient l x = match l with
2 | e::t when e=x-> true
3 | e::t-> contient t x;;
```

```
3.
1 let rec contient l x = match l with
2 | [e] -> e=x
3 | e::t when e=x-> true
4 | e::t-> contient t x;;
```

```
4.
1 let rec contient l x = match l with
2 | [] -> false
3 | e::t when e=x-> true
4 | e::t-> contient t x;;
```

Exercice

Exercice de cours 2.

1. Rappelez ce que signifie O , θ , Ω .
2. Quelle est la complexité en temps du tri rapide? Preuve attendue (méthode en cascade).

3. Si vous n'arrivez pas à faire la preuve (shame on you) : Quelle est la différence entre complexité amortie et complexité moyenne ?

1.2 Exercices intermédiaires

Exercice

Comparaison de complexité d'algorithmes.

Dans chacun des cas, dire si il vaut mieux avoir un algorithme en complexité 1 ou 2, si les deux reviennent au même, ou si on ne peut rien dire.

1. $O(\sum_{i=1}^n i)$ ou $O(n^2)$?
2. $O(\sum_{i=1}^n i^2)$ ou $O(n^\pi)$?
3. $O(n^{\log(n)})$ ou $O(\log(n)^n)$?
4. $O(e^n)$ ou $O(n!)$?

Exercice

Considération sur les notations de Landrau

Prouver ou contre-prouver :

1. $\forall f, g. f(n) + g(n) = \theta(\min(f(n), g(n)))$
2. $\forall (a, b) \in \mathbb{R}^2. (n + a)^b = \theta(n^b)$

Exercice

Relation récursive de complexité.

Résoudre $T(n) = 2T(\lfloor n/2 \rfloor) + n$.

1.3 Exercices plus compliqués

Exercice

Stirling.

Montrer que $n! = o(n^n)$.

Exercice

Stooge Sort.

Considérer l'algorithme suivant sur un tableau T et un entier n :

- If $n = 2$ and $T[0] > T[1]$, échanger $T[0]$ et $T[1]$;
- If $n \geq 3$:
 - Trier récursivement les premiers $\lceil \frac{2}{3}n \rceil$ éléments,

- Trier récursivement les derniers $\lceil \frac{2}{3}n \rceil$ éléments,
- Trier récursivement les premiers $\lceil \frac{2}{3}n \rceil$ éléments,

Après avoir lu l'algorithme, répondez aux questions suivantes :

- Montrer que l'algorithme est correcte.
- En donner la complexité dans le pire cas.
- Est-ce un bon tri ?

Exercice

Deux piles efficaces.

Expliquer comment implémenter une file avec deux piles en maintenant un **temps amorti** $O(1)$ par opérations.