

## Complexité des algorithmes

### Correction de l'Examen du 25 mai 2022

#### Exercice 1 :

(6 pts)

- $f_1 \in O(4^{n^4})$  Exponentiel
  - $f_2 \in O(n^3 \log n)$  Polynomial
  - $f_3 \in O(n \log n)$  Polynomial
  - $f_4 \in O(n^2)$  Polynomial
  - $f_5 \in O(5^n)$  Exponentiel
  - $f_6 \in O(\log n)$  Logarithmique
- $f_6 \ll f_3 \ll f_4 \ll f_2 \ll f_5 \ll f_1$

**Exercice 2 :** On considèrera uniquement les opérations d'affectation. Calculer le coût de cet algorithme.

**Q1 :** Algorithmes pour les deux représentations et leurs coûts.

(4.5 pts)

R1 : représentation usuelle

R2 : représentation par un tableau de triplets  $(i, j, a \neq 0)$

**Appel :** AlgR1

**Données :** une matrice  $n \times n$

**Résultat :** un entier

**Début**

$i, j, max$  : entiers

$max \leftarrow 0$

**pour**  $i \leftarrow 0$  à  $n - 1$  **faire**

**pour**  $j \leftarrow 0$  à  $n - 1$  **faire**

**si**  $max < A[i, j]$  **alors**

$max \leftarrow A[i, j]$

**fin**

**fin**

**fin**

**retourner**  $max$

**Fin**

**Appel :** AlgR2

**Données :** un tableau de triplets de taille  $p$

**Résultat :** un entier

**Début**

$i, max$  : entiers

$max \leftarrow 0$

**pour**  $i \leftarrow 0$  à  $p$  **faire**

**si**  $max < T[i].a$  **alors**

$max \leftarrow T[i].a$

**fin**

**fin**

**retourner**  $max$

**Fin**

**Complexité temporelle (nombre de comparaisons) :**

$$C_{AlgR1}(n) = n^2$$

$$C_{AlgR2}(p) = p$$

**Q2 :** Comparaison des complexités spatiales des deux représentations.

(4.5 pts)

$$E_{R1}(n) = n^2$$

$$E_{R2}(p) = 3p$$

La valeur critique du nombre d'éléments non nuls ( $p$ ) à partir de laquelle une méthode l'emporte sur l'autre :

$$p < \frac{n^2}{3}$$

**Question de TP :**

**Principe du tri par fusion**

(2.5 pts)

Le tri-fusion se fonde sur la stratégie "diviser pour régner".

1. On **divise** le tableau de départ de  $n$  éléments en deux sous-tableaux, ayant respectivement  $\lfloor n/2 \rfloor$  et  $\lceil n/2 \rceil$  éléments.
2. Un **appel récursif** est réalisé pour chacun de sous-tableaux.
3. On produit un tableau trié en **interclassant** tous les éléments des deux sous-tableaux triés.

**Equation récursive de complexité du tri par fusion**

(2.5 pts)

$$C_{TF}(n) = 2 \times C_{TF}(n/2) + C_{InterClass}(n)$$