UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

# DEVOIR 1

PAR

JÉRÉMY BOUCHARD (BOUJ08019605)

ALEXANDRE LAROUCHE (LARA04119705)

JEAN-PHILIPPE SAVARD (SAVJ04079609)

ALEXIS VALOTAIRE (VALA)

DEVOIR PRÉSENTÉ À

M. FRANÇOIS LEMIEUX

DANS LE CADRE DU COURS D'ALGORITHMIQUE (8INF433)

# Question 1

(a)

$$4^{log_2 n} \in \omega 4^{log_4 n}$$

(b)

$$(n+1)^2 \in \Theta(n^2)$$

(c)

$$2^{n+1} \in \Theta(2^n)$$

(d)

$$n! \in o(n+1)!$$

(e)

$$2^n \in \Theta(4^{n/2})$$

(f)

$$nlog_2 n \in o(n^2 - n)$$

(g)

$$(log_2 n)^2 \in \omega(log_2 n^2)$$

(h)

$$(1)^n \in o(1)$$

(i)

$$2^1 00 \in \Theta(2)$$

(j)

$$log_2 n^2 \in \Theta log_4 n$$

# Question 2

## a)

Il est possible d'exprimer le pseudo-code de l'algorithme décrit dans l'énoncé de la question 2. Il est également possible de constater qu'il s'agit de l'algorithme du tri par selection. Celui-ci ce décrit de cette façon :

**Algorithme 1 :** Pseudo-code du numéro 2

**Données :** $T[1 \cdots n]$
**Résultat :** $T[1 \cdots n]$ (Trié)

**pour** $j \leftarrow 1$ **à** $n-1$ **faire**
  $petit \leftarrow j$
  **pour** $i \leftarrow j+1$ **à** $n$ **faire**
    **si** $T[i] < T[petit]$ **alors**
      $petit \leftarrow i$
    **fin**
  **fin**
  $T[j] \leftrightarrow T[petit]$
**fin**

## b)

Il est possible de calculer le temps d'exécution de cette algorithme facilement. En effet, pour chaque élément du tableau il est necessaire de parcourir le reste du tableau moins les éléments déjà triés. Donc, pour la première itération, il est nécessaire de réaliser $(n-1)$ comparaisons. Pour la deuxième itération, cela sera $(n-2)$ itérations et ainsi de suite jusqu'à ce qu'il ne reste que une seule valeur à trier. Il est donc possible de formaliser le temps d'exécution de cette manière :

$$(n-1) + (n-2) + (n-3) + ... + 1 = \sum_{i=1}^{n-1} i$$

Il ne reste qu'à trouver l'ordre de $\sum_{i=1}^{n-1} i$.

*Temps d'exécution de l'algorithme de tri.*

$$
\begin{aligned}
S &= \sum_{i=1}^{n-1} i \\
&= (n-1) + (n-2) + (n-3) + ... + 1 \\
2S &= (n-1) + (n-2) + (n-3) + ... + 1 \\
&\qquad + 1 \qquad\quad + 2 \qquad\quad + 3 \; + ... + (n-1) \\
&= \underbrace{(n + n + n + ... + n)}_{(n-1) \text{ fois}} \\
&= (n-1)(n) \\
S &= \frac{n(n-1)}{2} \\
&= \frac{1}{2}n^2 - \frac{1}{2}n
\end{aligned}
$$

*D'où* $\mathcal{O}(n^2)$ $\square$

# Question 3

Supposons, par abus de notation, que $f \geq g$ (puisque forcément l'une doit être plus grande ou égale que l'autre).

## a)

Prouvons que $max(f, g) \in \mathcal{O}(f + g)$:
On a

$$max(f, g) = f$$

Or,

$$f \leq f + g$$

On a donc

$$max(f, g) \in \mathcal{O}(f + g)$$

## b)

Prouvons que $f + g \in \mathcal{O}(max(f, g))$:
On a

$$f + g \in \mathcal{O}(f)$$

Or,

$$max(f, g) = f$$

Alors,

$$f + g \in \mathcal{O}(max(f, g))$$

Donc,

$$\mathcal{O}(max(f, g)) = \mathcal{O}(f + g)$$

CQFD

## Question 4

## Question 5