Université Savoie Mont Blanc,
UFR SceM
INFO602, L3 INFO

March 17, 2020

TD 1, Notations $\mathcal{O}$,Θ, et Ω et analyse en pire cas

---

(1) Transitivité de $\mathcal{O}$ $f(n) = \mathcal{O}(g(n))$, et $g(n) = \mathcal{O}(h(n))$, implique $f(n) = \mathcal{O}(h(n))$

(2) Transitivité de Ω $f(n) = \Omega(g(n))$, et $g(n) = \Omega(h(n))$, implique $f(n) = \Omega(h(n))$

(3) Transitivité de Θ $f(n) = \Theta(g(n))$, et $g(n) = \Theta(h(n))$, implique $f(n) = \Theta(h(n))$

(4) Règle des sommes $f(n) + g(n) = \mathcal{O}(\max(f(n), g(n))$

(5) Règle des sommes $f(n) + g(n) = \Omega(\min(f(n), g(n))$

(6) Règle des produits $f(n)\mathcal{O}(g(n)) = \mathcal{O}(f(n)g(n)$, vrai pour Θ et Ω aussi.

(7) Polynômes $n^a = \mathcal{O}(n^b)$ lorsque $0 \leq a \leq b$.

(8) Polynômes $a_0 + a_1 n + a_2 n^2 + \cdots + a_k n^k = \Theta(n^k)$ lorsque $a_k > 0$ (seul le monôme de plus grand degré compte dans la complexité).

(9) Equ. logarithmes $\log_b n = \Theta(\ln n)$, on écrira donc souvent log sans préciser.

(10) Logarithmes $\log n = \Omega(1)$ et $\log n = \mathcal{O}(n^\epsilon)$, pour tout $\epsilon > 0$ (log est dominé par les polynômes non constants)

(11) Exponentielles $a^n = \mathcal{O}(b^n)$, pour $0 < a \leq b$

(12) Exponentielles $e^n = \Omega(n^k)$ pour n'importe quel $k > 0$ (exp domine les polynômes).

Exercice 1. Notations O et Ω

Indiquez si les relations suivantes sont justes ou fausses. Pour chaque question n est un nombre que l'on fait tendre vers l'infini. Précisez quelle(s) relation(s) vous avez utilisé(es).

1. $4n^2 + 2n \log n = O(n^2)$
2. $2^n = O(n^2)$
3. $n + 3n \log n = O(n \log n)$
4. $\sqrt{n} \log n = O(n^{\frac{2}{3}})$
5. $\log n = O(\sqrt{n})$.
6. $2^n = \Omega(n^3)$
7. $n^2 - 2n^3 + \frac{1}{4}n^4 = \Omega(n^4)$
8. $\frac{3n}{\log n} = \Omega(n)$.

Exercice 2. Notations O et Ω

Complétez le tableau ci-dessous en indiquant si f est un grand O, un grand Ω ou un grand Θ de la fonction g (pour n tendant vers l'infini).

		g			
		$n^3 + 2n^2 \log n$	$3n^2 + 4$	$n \log n + 12n$	$n\sqrt{n} + n \log n$
f	$27 \log n + 10$				
	$4n^2 + n$				
	$(n+5) \log n$				
	$(3/2)^n$				

Vous pourrez aussi expliquer vos résultats.

Exercice 3. Utilisation des dérivées pour montrer des relations asymptotiques

Montrez que si, pour tout $x \geq x_0$, $f'(x) \leq g'(x)$ et $f(x) > a$, où a est une constante strictement positive, alors $f(n) = \mathcal{O}(g(n))$.

Exercice 4. Domination d'exponentielle sur n'importe quel polynôme

Montrez que $n^k = O(e^n)$ pour tout k entier. On utilisera l'exercice précédent (exo 2).

Exercice 5. Equivalence des logarithmes

Pourquoi $\log_b n = \Theta(\ln n)$?

Exercice 6. Equivalence des exponentielles

Montrez que :

- $a^n = \Theta(b^n)$ si $a = b$
- $a^n = \Theta(b^n)$ est faux sinon

Exercice 7. Logarithmes

Montrez que $\log n = \Omega(\log \log n)$ mais que $\log n = \mathcal{O}(\log \log n)$ est faux. Ils ne sont donc pas dans la même classe de complexité.

En revanche, on a bien $\log n = \Omega(1)$ et $\log \log n = \Omega(1)$.

Exercice 8. Complexité de argument minimum

Quelle est la complexité du code ci-dessous en fonction des paramètres ? Plus généralement, quelle est sa complexité en pire cas si le tableau T contient n entiers au maximum ?

```
/* ArgMin ou argument minimum : indice de l'élément dont la valeur est plus petite que les
  valeurs de tous les autres. */
Fonction ArgMin(E T : TabEntier, E i, j : entier) : entier ;
Var : k,m : entier ;
début
  m ←i ;
  Pour k de i + 1 à j Faire
    si T[k] < T[m] alors m ←k;
  Retourner m;
```

Exercice 9. Complexité de tri insertion

Quelle est la complexité du tri insertion donné ci-dessous en fonction des paramètres ? Est-ce $\Theta(n^2)$?

Exercice 10. Complexité du calcul des coefficients binomiaux par tableau

Quelle est la complexité de l'algorithme de calcul du coefficient binomial C_n^k avec un tableau comme dans l'algorithme ci-dessous ?

Optimisez-le afin que sa complexité soit seulement de $\Theta(kn)$.

Enfin, on sait par ailleurs que $C_n^k = \frac{n!}{(n-k)!k!}$. Est-ce que le calcul avec la factorielle serait plus rapide ?

```
Action TriInsertion(ES T : TabEntier, E n : entier);
Var : i,j : entier ;
début
    Pour i de 1 à n − 1 Faire
        /* les éléments de 0 à i-1 sont triés. */;
        j ← i − 1 ;
        Tant Que j >= 0 et T[j + 1] < T[j] Faire
            début
                Echange(T[j], T[j + 1]);
                j ← j − 1;
```

```
Action Binomial(E n, k : entier );
Var : i,j : entier ;
T : Tableau[0 . . . n] d'entiers
début
    T[0] ← 1 ;
    Pour j de 1 à n Faire
        T[j] ← 0
    Pour i de 1 à n Faire
        Pour j de n à 1 par pas de −1 Faire
            T[i] ← T[i − 1] + T[i]
    Retourner T[k];
```