Université Assane Seck de Ziguinchor
UFR sciences et Technologies
Département Informatique

Exercices de TD (Feuille 3)

Consignes : Les exercices mis en évidence sont obligatoires pour valider le chapitre 3. Les autres exercices sont optionnels. Il vous est conseillé de les faire après les exercices obligatoires.

3.1.1 Soient $f(n)$ et $g(n)$ des fonctions asymptotiquement non négatives. En s'aidant de la définition de base de la notation $\boldsymbol{\Theta}$, prouver que $\max(f(n), g(n)) = \boldsymbol{\Theta}(f(n) + g(n))$.

3.1.2 Montrer que, pour deux constantes réelles a et b quelconques avec $b > 0$, l'on a
$(n + a)^b = \boldsymbol{\Theta}(n^b)$. (3.2)

3.1.3 Expliquer pourquoi l'affirmation « Le temps d'exécution de l'algorithme *A* est au moins $O(n^2)$ » n'a pas de sens.

3.1.4 Est-ce que $2^{n+1} = O(2^n)$? Est-ce que $2^{2n} = \boldsymbol{\Theta}(2^n)$?

3.1.5 Démontrer le théorème 3.1.

3.1.6 Démontrer que le temps d'exécution d'un algorithme est $\boldsymbol{\Theta}(g(n))$ si et seulement si son temps d'exécution dans le cas le plus défavorable est $O(g(n))$ et son temps d'exécution dans le meilleur des cas est $\boldsymbol{\Omega}(g(n))$.

3.1.7 Démontrer que $o(g(n)) \cap \omega(g(n))$ est l'ensemble vide.

3.2.1 Montrer que, si $f(n)$ et $g(n)$ sont des fonctions monotones croissantes, alors les fonctions $f(n) + g(n)$ et $f(g(n))$ le sont également ; montrer que, si $f(n)$ et $g(n)$ sont en outre non négatives, $f(n) \cdot g(n)$ est monotone croissante.

3.2.2 Démontrer l'équation (3.15).

3.2.3 Prouver l'équation (3.18). Montrer aussi que $n! = \boldsymbol{\omega}(2^n)$ et $n! = o(n^n)$.

3.2.5 Laquelle de ces deux fonctions est la plus grande asymptotiquement : $\lg(\lg^* n)$ ou $\lg^*(\lg n)$?

3.2.6 Démontrer par récurrence que le *i*-ème nombre de Fibonacci satisfait à l'égalité