ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Sections d'Informatique et de Systèmes de Communication

Série d'exercices 2 28 Septembre 2007

1. Induction

a) Prouver par induction que pour tout $n \in \mathbb{N}$, $x \geq -1$, on a

$$(1+x)^n \geq 1 + nx.$$

b) Prouver par induction que pour tout $n \in \mathbb{N}$, on a

$$2(\sqrt{n+1} - 1) < 1 + \frac{1}{\sqrt{2}} + \ldots + \frac{1}{\sqrt{n}} < 2\sqrt{n}.$$

2. Relations de récurrence et nombres de Fibonacci

Nous considérons les suites $(x_1, x_2, \ldots, x_n)$ d'éléments de $\{0, 1\}$. Il y a 2^n suites de ce genre, mais nous ne nous intéressons qu'à celles dans lesquelles il n'y a jamais deux zéros consécutifs.

Par exemple avec $n = 3$, l'ensemble de toutes les suites possibles est $\{000, 001, 010, 011, 100, 101, 110, 111\}$, mais l'ensemble des suites qui nous intéressent est $\{010, 011, 101, 110, 111\}$.

Soit $S(n)$ l'ensemble des suites de taille n vérifiant cette propriété. On pose $F(n) = |S(n)|$.

a) Quels sont les ensembles $S(1)$, $S(2)$, $S(3)$ et $S(4)$? Montrer que $F(1) = 2$, $F(2) = 3$, $F(3) = 5$, et $F(4) = 8$.

b) Considérons $S(n)$. Montrer que

$$\Big|\, \{(x_1, \ldots x_n) \in S(n) \mid x_n = 1\} \,\Big| = |S(n-1)| = F(n-1),$$

et que

$$\Big|\, \{(x_1, \ldots x_n) \in S(n) \mid x_n = 0\} \,\Big| = |S(n-2)| = F(n-2).$$

c) Prouver par induction que :

$$F(n) = F(n-1) + F(n-2)$$

3. La notation O

a) Montrer que $n^2 + 100 = O(n^2)$, c'est à dire trouver des éléments c et n_0 satisfaisant la définition. Montrer maintenant que $n^2 + 100 = \theta(n^2)$.

b) Supposons que $f(n) = O(g(n))$ et $g(n) = O(h(n))$. Montrer que $f(n) = O(h(n))$.

c) Montrer que $n = \Omega(\log_2(n))$.

d) Supposons que $f(n) = O(g(n))$. Montrer que pour tout $a, b \in \mathbb{R}_{>0}$ on a $a \cdot f(n) = O(b \cdot g(n))$

e) Montrer que pour tout $d \in \mathbb{N}$, et pour tout $a \in \mathbb{R}$ avec $a > 1$, on a $n^d = O(a^n)$. (Indice : prendre le logarithme des deux fonctions).

f) Remplir le tableau suivant par vrai ou faux.

$f(n)$	$g(n)$	$f(n) = O(g(n))$	$f(n) = \Omega(g(n))$	$f(n) = \theta(g(n))$	$f(n) = o(g(n))$
$n^{1/100}$	$\sqrt{n}$				
$\ln(n)$	$\ln^2(n)$				
$\sqrt{n}$	$\ln^2(n)$				
2^n	$n!$				
$\log_2(n)$	$\log_3(n)$				
$\ln(n)$	$\ln\ln(n)$				
$2^{\ln(n)}$	n^2				
2^n	$n^{\ln\ln(n)}$				
$2^{\sqrt{\ln(n)}}$	$\sqrt{n}$				

4. Relations de récurrence

a) Soit $T : \mathbb{N} \to \mathbb{R}$ une fonction monotone croissante (c'est-à-dire que $T(n) \leq T(n+1)$) telle que pour tout $n \in \mathbb{N}$ on a

$$T(10n) = 3 \cdot T(n) + 10^3 \cdot \sqrt{n}$$

Montrer que $T(n) = O(\sqrt{n})$.

b) Soit $T : \mathbb{N} \to \mathbb{R}$ une fonction monotone croissante telle que $T(1) = 1$, et pour tout $n \in \mathbb{N}$ on a

$$T(2n) = n \cdot T(n)$$

Nous voulons montrer que $T(n) = O(n^{\frac{\log_2(n)+1}{2}})$.

– (i) Soit n une puissance de 2, disons $n = 2^k$. Trouver une formule pour $T(n)$ en fonction de k.

– (ii) En déduire que pour tout $n \in \mathbb{N}$ on a

$$T(n) \leq 2^{\frac{\log_2 n \cdot (\log_2(n)+1)}{2}}$$

(utiliser le fait que $T(n) \leq T(2^{k+1})$, où $k = \lfloor \log_2 n \rfloor$ ainsi que la partie (i)).

– (iii) En déduire que $T(n) = O(n^{\frac{\log_2(n)+1}{2}})$ (il s'agit simplement d'utiliser des manipulations algébriques pour arriver au résultat).