IFT2015 A16 :: Examen intra[1]

Miklós Csűrös

14 octobre 2013

L'EXAMEN vaut 100 points, et vous pouvez avoir jusqu'à 15 points de boni additionnels.

- ★ Aucune documentation n'est permise.
- ★ Décrivez vos algorithmes en pseudocode ou en Java(-esque).
- ★ Répondez à toutes les questions dans les cahiers d'examen.

Exo	points	boni
F0	1	
F1	$6+7\times 2=20$	
F2	$3+12=15$	
F3	14	5
F4	$15+10=25$	5
F5	$10+10+5=25$	5
Σ	100	15

[1] The English translation follows.

F0 Votre nom (1 point)

▶ Mettez votre nom et code permanent sur tous les cahiers soumis.

F1 Théorie et pratique (20 points)

i. Presque partout (6 points) Soit $P(n)$ une propriété[2] des entiers naturels qui est soit vrai soit faux pour chaque $n = 0,1,2,\ldots$. ▶ Donnez une définition précise de l'expression "pour presque tout" dans l'énoncé $\boxed{P(n)\ \textit{est vrai}\text{ pour presque tout } n.}$

ii. Comparaisons de taux de croissance (14 points) ▶ Comparez le taux de croissance des fonctions dans les rangées. Pour chaque paire f,g, écrivez "=" si $f=\Theta(g)$, "$\ll$" si $f=o(g)$, "$\gg$" si $g=o(f)$, et "???" si aucun des trois cas n'applique. Chaque réponse vaut 2 points, et il n'est pas nécessaire de les justifier. $\lg n$ dénote le logarithme binaire de n.

[2] *Propriété* est une fonction logiq... $P\colon \{0,1,2,\ldots\} \mapsto \{\text{vrai}, \text{faux}\}$... exemple, $P(n) = \{f(n) \le cn\}$ ... constante $c > 0$.

★ Aucune documentation n'est permise.

★ Décrivez vos algorithmes en pseudocode ou en Java(-esque).

★ Répondez à toutes les questions dans les cahiers d'examen.

F2	$3+12=15$	
F3	14	5
F4	$15+10=25$	5
F5	$10+10+5=25$	5
Σ	100	15

F0 Votre nom (1 point)

▶ Mettez votre nom et code permanent sur tous les cahiers soumis.

F1 Théorie et pratique (20 points)

i. Presque partout (6 points) Soit $P(n)$ une propriété[2] des entiers naturels qui est soit vrai soit faux pour chaque $n = 0, 1, 2, \ldots,$. ▶ Donnez une définition précise de l'expression "pour presque tout" dans l'énoncé $\boxed{P(n) \textit{ est vrai } \text{pour presque tout } n.}$

ii. Comparaisons de taux de croissance (14 points) ▶ Comparez le taux de croissance des fonctions dans les rangées. Pour chaque paire f, g, écrivez "=" si $f = \Theta(g)$, "$\ll$" si $f = o(g)$, "$\gg$" si $g = o(f)$, et "???" si aucun des trois cas n'applique. Chaque réponse vaut 2 points, et il n'est pas nécessaire de les justifier. $\lg n$ dénote le logarithme binaire de n.

a	$f(n) = n^2 \cdot 2^{2015}$	$g(n) = (n+2015)^2$
b	$f(n) = \sqrt{\lg n}$	$g(n) = \lg(\sqrt{n})$
c	$f(n) = \sum_{i=0}^{n} 2015^i$	$g(n) = 2015^n$
d	$f(n) = \log_{2015}(n!)$	$g(n) = n \ln n$
e	$f(n) = 3 \lg n$	$g(n) = \log_{2015}(n)$
f	$f(n) = n \lg\lg(n+2)$	$g(n) = n(\lg(n+2))^2$
g	$f(n) = sin^2 n$	$g(n) = cos^2 n$

[2] *Propriété* est une fonction logique : $P\colon \{0,1,2,\ldots\} \mapsto \{\text{vrai}, \text{faux}\}$. Par exemple, $P(n) = \{f(n) \leq cn\}$ avec une constante $c > 0$.

g $f(n) = sin^2 n$ $\qquad$ $g(n) = cos^2 n$

F2 Types abstraits (15)

i. Œufs de coucou (3 points) ▶ Identifiez les trois types abstraits sur la liste suivante :

dictionnaire, Drake, liste chaînée, pile, queue (file FIFO), tableau trié.

ii. Interface (12 points) ▶ Spécifiez et décrivez[3] les opérations principales pour les trois types abstraits identifiés.

FIG. 1: Le rappeur Drake égaré dans une boîte ouverte.

[3] sans entrer dans les détails d'une implémentation quelconque !

F3 Chaîne Fibonacci (14+5 points)

Les *chaînes Fibonacci* f_n sont des chaînes de caractères $\boxed{0}$, $\boxed{1}$, définies par induction de la manière suivante. On a $f_0 = \boxed{0}$. Pour $n > 0$, f_n est dérivée à

FIG. 2: Leonardo

F3 *Chaîne Fibonacci (14+5 points)*

Les *chaînes Fibonacci* f_n sont des chaînes de caractères $\boxed{0}, \boxed{1}$, définies par induction de la manière suivante. On a $f_0 = \boxed{0}$. Pour $n > 0$, f_n est dérivée à partir de f_{n-1} en remplaçant chaque caractère en même temps selon les règles

$$\boxed{0} \xrightarrow{\text{remplacer par}} \boxed{1}$$
$$\boxed{1} \xrightarrow{\text{remplacer par}} \boxed{1}\boxed{0}$$

i. Remplacements (14 points) Dans cet exercice, on représente f_n comme une liste (simplement) chaînée sur laquelle chaque nœud[4] contient un symbole $\boxed{1}$ ou $\boxed{0}$. On veut un algorithme qui performe les substitutions sur une liste chaînée. On peut alors construire f_n en l'appellant n fois.

▶ Décrivez un algorithme, nommé FiboSubst(n), qui construit une nouvelle chaîne en exécutant les règles de substitution nœud par nœud à partir de x, créant des nœuds comme nécessaire pour la nouvelle liste. L'algorithme retourne la tête de la liste modifiée (la tête contient le premier caractère).

ii. Fibonacci ? (5 points boni) ▶ Démontrez que $f_n = f_{n-1} \oplus f_{n-2}$ pour chaque $n > 1$, où $\oplus$ dénote la concaténation.

F4 *Croissance sous-exponentielle (25+5 points)*

FIG. 2: Leonardo Fibonacci (1175–1250)

TAB. 1: Chaînes Fibonacci

$f_0 = \boxed{0}$

$f_1 = \boxed{1}$

$f_2 = \boxed{1}\boxed{0}$

$f_3 = \boxed{1}\boxed{0}\boxed{1}$

$f_4 = \boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{0}$

$f_5 = \boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{1}$

[4] Chaque nœud x (sauf le nœud terminal x = null) contient les variables x.next (prochain élément) et x.val $= \boxed{0}, \boxed{1}$. Pour créer (instancier) un nouveau nœud, utilisez l'opération prédéfinie newNode(c) avec $c = \boxed{0}, \boxed{1}$.

```
FiboChain(n)                    // construit la chaîne f_n
  x ← newNode([0])              // f_0
  for i ← 1,...,n {x ← FiboSubst(x)}
  return x
```

chaque $n > 1$, où $\oplus$ dénote la concaténation.

```
x ← newNode( [0] )                      // f_0
for i ← 1,...,n {x ← FiboSubst(x)}
return x
```

FIG. 3: FiboChain(4) retourne la tête de la liste chaînée illustrée ici.

F4 *Croissance sous-exponentielle (25+5 points)*

Dans cet exercice, on étudie les notions de sous-exponentielle et super-polynomiale, définies ainsi pour une fonction $f(n)$ sur $n = 0, 1, 2, \ldots$:

$$f \text{ est } \textit{sous-exponentielle} \text{ si } \qquad f(n) = 2^{o(n)} \qquad \text{(F1a)}$$

$$f \text{ est } \textit{super-polynomiale} \text{ si } \qquad n^{O(1)} = o\big(f(n)\big) \qquad \text{(F1b)}$$

i. (15 points) ▶ Donnez des définitions équivalentes[5] à (F1a) et (F1b) sans utiliser la notation asymptotique $(o, O, \Theta, \Omega, \ldots)$.

ii. (10 points) ▶ Démontrez[6] que la fonction

$$g(n) = \begin{cases} n+1 & \{n = 0, 1, \ldots, 2014\} \\ n^{n/\log_{2015} n} & \{n \geq 2015\} \end{cases}$$

est super-polynomiale mais non pas sous-exponentielle.

iii. (5 points boni) Est-ce qu'il existe une fonction qui est super-polynomiale et sous-exponentielle en même temps ? ▶ Donnez un exemple, ou démontrez que c'est impossible.

[5] ***Indice:*** Découvrez la relation entre $f(n)$ et les fonctions cachées par les termes asymptotiques, et exprimez la contrainte imposée soit comme une inégalité qui vaut presque partout, soit — si applique — comme une limite $\lim_{n\to\infty}$.

[6] ***Indice:*** Simplifiez $n^{n/\log_{2105} n}$.