Faculté des sciences Dhar El-Mehraz, Fès — Session normale 2018/2019

Contrôle Final : SMI - Semestre S3 — **Module : Algorithmique II**

Durée : 1h30

Exercice 1 : (2 pts)

Écrire une procédure (i.e. fonction sans valeur de retour) qui prend en entrée un entier naturel *a* et le transforme écrit à l'envers. Par exemple, pour *a=1234*, un appel à la procédure demandée, doit le transformer en *a=4321*, de manière persistante.

Exercice 2 : (4 pts)

En supposant qu'on ne sait faire que des opérations d'addition et de soustraction, il vous est demandé d'écrire deux fonctions *quotient(n,p)* et *reste(n,p)* qui retournent respectivement le quotient et le reste de la division entière (*n/p*).

1- Écrire la fonction *reste(n,p)* qui retourne le reste de la division (*n/p*).

2- Écrire la fonction *quotient(n,p)* qui retourne le quotient de la division (*n/p*).

Exercice 3 : (6 pts)

1- Écrire une fonction *divise(p,q)* d'argument deux entiers naturels non nuls *p* et *q* et renvoyant VRAI si *p* divise *q*, et FAUX sinon.

2- Écrire une fonction *estPremier(p)* d'argument un entier naturel *p*, renvoyant *1* si *p* est premier, et renvoyant *0* sinon.

3- Écrire une fonction *nbPremiers(n)* d'argument un entier naturel *n* et renvoyant le nombre de nombres premiers inférieurs ou égaux à *n*.

Exercice 4 : (8 pts)

1- Écrire une fonction récursive permettant de trier dans le sens croissant, un tableau de valeurs réelles par la méthode de *tri à bulles*.

2- Écrire les fonctions permettant de trier le sens croissant, un tableau de valeurs réelles par la méthode de *tri par fusion*.

3- Analyser en détails, les complexités maximales, en termes du nombre d'échanges, de chacune des deux fonctions de tri établies aux questions 1 et 2.

Faculté des sciences Dhar El-Mehraz, Fès | Session rattrapage 2018/2019

Contrôle Final : SMI - Semestre S3 **Module : Algorithmique II**

Durée : 1h30

Exercice 1 : (4 pts)

Énoncé

On rappelle que si n, k sont deux entiers naturels, le coefficient binomial $\binom{n}{k}$ vérifie

- $\binom{n}{k} = 0$ si $k > n$.
- $\binom{n}{0} = \binom{n}{n} = 1$.
- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ si $1 \leq k \leq n-1$.

Écrire une fonction récursive permettant de calculer $\binom{n}{k}$ à partir des formules précédentes.

Exercice 2 : (4 pts)

Énoncé

Soit $P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$. Pour évaluer $P(x)$, le mathématicien anglais Horner a proposé la méthode suivante

$$P(x) = (\cdots(((a_n x + a_{n-1})x + a_{n-2})x + a_{n-3})\cdots)x + a_0.$$

Écrire une fonction qui prend en entrée la liste des coefficients d'un polynôme P et un nombre réel x et qui retourne $P(x)$ suivant la méthode de Horner. La taille de la liste P doit aussi être passée comme argument.

Exercice 3 : (6 pts)

Énoncé

L'algorithme d'exponentiation rapide est basé sur la remarque suivante : on a $a^{2p} = a^p \cdot a^p$ et $a^{2p+1} = a^p \cdot a^p \cdot a$. Ainsi, pour calculer a^{2p}, il suffit de savoir calculer a^p et de faire une multiplication, et pour calculer a^{2p+1}, il suffit de savoir calculer a^p et de faire deux multiplications.

1. Ecrire une fonction récursive implémentant cet algorithme.
   fonction exporapide(a: reel, n :entier) : reel
2. Démontrer que, la complexité maximale en termes du nombre total de multiplications effectuées par un appel à exporapide(a,n) est logarithmique.

Exercice 4 : (6 pts)

Énoncé

1. Ecrire une fonction non récursive permettant de chercher, par dichotomie, une valeur donnée x dans un tableau trié de n valeurs réelles.

2. Détailler l'analyse de la complexité de cet algorithme, dans le pire des cas.