Université Lille1 Sciences et Technologies

Info 204 - ASD

2009–2010

DS1 - documents de cours, TD, TP autorisés

Note : lorsque, dans une question, est demandé la complexité asymptotique, la réponse attendue est soit une notation $\mathcal{O}$, soit un notation Θ soit une notation Ω.

Exercice 1 : Complexités asymptotiques

Indiquer pour chacune des égalités si elles sont vraies ou fausses ou si vous ne savez. **Attention, notation QCM : réponse fausse = des points en moins.**

1. $n^2 = \Omega(n)$
2. $n = \Omega(n^2)$
3. $\log n = \mathcal{O}(n^2)$
4. $\log n = \Theta(n^2)$

Exercice 2 : Récurrences

On dispose de deux algorithmes dont les complexités sont exprimées ainsi :

Algorithme 1 $c(n) = 2c(n-1) + 3c(n-2), c(1) = 4, c(0) = 0$

Algorithme 2 $c(n) = 2c(\frac{n}{4}) + 2\sqrt{n}$

Q 2.1 Quelle est la complexité asymotique de l'algorithme 1. Justifier clairement.

Q 2.2 Quelle est la complexité asymotique de l'algorithme 2. Justifier clairement.

Exercice 3 : Programmation dynamique

On rappelle ci-dessous la formule de Pascal pour le calcul des coefficients binomiaux :

$$\begin{cases} C_n^0 &= 1 \\ C_n^n &= 1 \\ C_n^p &= C_{n-1}^p + C_{n-1}^{p-1} \end{cases}$$

Q 3.1 Ecrire en PASCAL une fonction récursive de calcul de C_n^p utilisant cette formulation :

```
function coeff(n, p : CARDINAL) : CARDINAL;
```

Q 3.2 La fonction écrite à la question précédente est-elle récursive terminale ? Justifier clairement.

Q 3.3 On note $a(n,p)$ le nombre d'additions réalisées lors du calcul de C_n^p. Donner une formule récursive pour l'expression de $a(n,p)$ en fonction des $a(i,j), i \leq n, j \leq p$ (on ne demande pas de résoudre cette équation).

Q 3.4 Quelle est votre intuition quant à la complexité asymptotique de l'algorithme récursif ? Pourquoi ?

Le calcul de l'ensemble des C_n^p peut-être exécuté plus rapidement grâce au triangle de Pascal, qui calcule pour chaque n puis pour chaque p les valeurs de C_n^p en utilisant les calculs réalisés à la ligne précédente. Une illustration est donnée ci-dessous :

		p			
		0	1	2	3
	0	1			
	1	1	1		
n	2	1	2	1	
	3	1	3	3	1

On propose l'implantation suivante afin de réaliser le calcul des C_n^p par programmation dynamique :

```
1  // CU : 0 <= p <= n, 0 <= n
2  function progdyn (n : CARDINAL; p : CARDINAL) : CARDINAL;
3  var
4     i, j : CARDINAL;
5     t    : array of CARDINAL;
6  begin
7     setlength(t,n+1);
8     // initialisation
9     t[0] := 1;
10    // remplissage
11    for i := 1 to n do begin
12       t[i] := 1;
13       for j := i-1 downto 1 do
14          t[j] := t[j] + t[j-1];
15       afficher(t);
16    end {for};
17    // resultat
18    -- voir question 3.6
19 end {progdyn};
```

Q 3.5 Ecrire ce qui est affiché par la fonction pour $n = 5$ et $p = 3$.

Q 3.6 Donner l'instruction de la ligne 18.

Q 3.7 Exprimer le nombre d'additions en fonction de n et p. Donner une formule exacte.

Q 3.8 Quelle est la complexité asymtotique en temps de l'algorithme en fonction de n et p ? Et en espace ?

Q 3.9 Dans l'algorithme proposé, certaines cases de `t` sont calculées alors qu'elles sont inutiles pour l'obtention du résultat final. Proposer la (les) modification(s) nécessaire(s).

Q 3.10 Exprimer le nombre d'additions en fonction de n et p. Donner une formule exacte.