Université des Sciences et de la Technologie Houari Boumediène
Faculté d'Electronique et d'Informatique
Département d'Informatique
LMD Master 1ère Année IL 2013/2014
Module ''Algorithmique Avancée et Complexité''
Date : 16/12/2013 (par Mr Isli)

Corrigé de l'interrogation

Exercice 1 <5 points=2,5+2,5>:
Soit $f(n)=5n^2\log(n)+8n^2+5$ le nombre d'opérations élémentaires du pire cas d'un algorithme A.

1. Montrez que $f(n)=O(n^2\log(n))$.
2. A-t-on $f(n)=O(n.\log(n))$? Expliquez.

Solution :

1. Montrons que $f(n)=O(n^2\log(n))$. Il suffit de trouver $c>0$ et $n_0\geq 0$ tels que $\forall n\geq n_0$ $f(n)\leq c.n^2\log(n)$:
   $5n^2\log(n)+8n^2+5\leq c.n^2\log(n)$ on divise les deux membres par $n^2\log(n)$
   $5+\frac{8}{\log(n)}+\frac{5}{n^2\log(n)}\leq c$
   Prendre $c=14$ et $n_0=10$ ($5+\frac{8}{\log(n)}+\frac{5}{n^2\log(n)}$ décroissante à partir de $n=10$)
   Conclusion : $f(n)=O(n^2\log(n))$.
2. Supposons que $f(n)=O(n.\log(n))$. Il existerait $c>0$ et $n_0\geq 0$ tels que $\forall n\geq n_0$ $f(n)\leq c.n.\log(n)$:
   $5n^2\log(n)+8n^2+5\leq c.n\log(n)$ on divise les deux membres par $n\log(n)$
   $5n+\frac{8n}{\log(n)}+\frac{5}{n.\log(n)}\leq c$
   La constante c est inexistante vu que la limite du premier membre quand n tend vers l'infini est l'infini : d'où contradiction
   Conclusion : on n'a pas $f(n)=O(n.\log(n))$

Exercice 2 <4 points>:
Donnez un algorithme polynômial testant si une chaîne de caractères v est sous-chaîne d'une chaîne u ; c'est-à-dire s'il existe deux chaînes w et w' telles que u=wvw'. Calculez la complexité de l'algorithme.

Solution :
Sous-mot(v,u)
Début
1. n=longueur(u)
2. m=longueur(v)
3. borne= n-m+1
4. sousmot=faux
5. I=1
6. Tant que (non sousmot) et i≤borne faire
   a. J=1
   b. Tant que j≤m et u[i+j-1]=v[j] faire
      i. j=j+1
   c. fait
   d. Si j=m+1 alors sousmot=vrai finsi
7. Fait
8. retourner sousmot

Fin

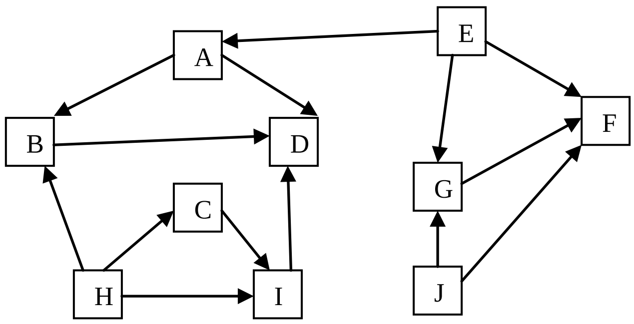
Le nombre f(n,m) d'opérations élémentaires du pire cas de l'algorithme est donné par le tableau ci-dessous :

Instruction	**Nombre d'opérations du pire cas**
1.+2.+3.+4.+5.	7
6.	2(n-m+2)
6.a.	n-m+1
6.b.	4(n-m+1)(m+1)
6.b.i.	2(n-m+1)m
6.d.	2(n-m+1)+1
8.	1

$f(n,m)=7+2(n-m+2)+ n-m+1+4(n-m+1)(m+1)+ 2(n-m+1)m+2(n-m+1)+1+1$
$\quad =7+2n-2m+4+n-m+1+4nm+4-4m^2-4m+4m+4+2nm-2m^2+2m+2n-2m+2+2$
$\quad =24+5n-3m+6nm-6m^2$

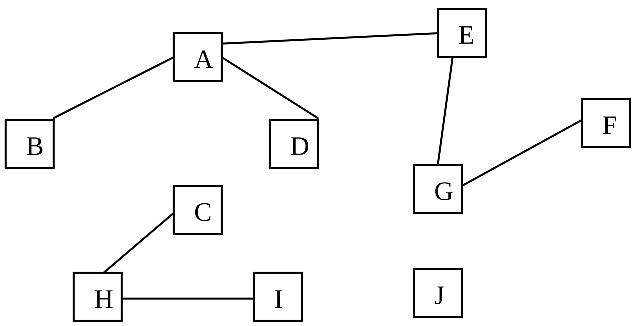
Conclusion : $f(n,m)=O([\max(n,m)]^2)$ (complexité quadratique)

Exercice 3 <5 points=3+2>:
On considère le graphe orienté suivant :



1. Donnez une forêt de recouvrement issue d'un parcours en profondeur d'abord du graphe
2. Donnez l'ordre dans lequel le parcours considéré a visité les sommets du graphe

Solution :
1. La forêt de recouvrement ci-dessous est issue d'un parcours en profondeur d'abord du graphe, et est constituée de trois arbres dont les racines sont E, H et J



2. Le parcours en profondeur d'abord dont est issue la forêt de recouvrement ci-dessus a visité les sommets dans l'ordre EADBGFHICJ

Remarque : d'autres solutions existent !

Exercice 4 <6 points=1+1+2+1+1>:

On considère le problème de décision P suivant :

- **Description :** deux entiers strictement positifs a et b
- **Question :** existe-t-il un entier x tel que x^2=a mod b ?

Le but de l'exercice est de trouver un algorithme polynômial de validation pour le problème P ci-dessus en procédant comme suit :

1. Donnez une structure de données permettant de représenter une instance du problème P. Expliquez
2. En quoi consiste un certificat d'une instance du problème P. Donnez une structure de données permettant la représentation d'un tel certificat. Expliquez
3. Donnez un algorithme de validation pour le problème P sous forme d'une fonction booléenne dont il est important que vous expliquiez les paramètres. L'algorithme, que vous appellerez validation_P, doit évidemment être polynômial, la preuve de la polynômialité faisant l'objet des questions 4 et 5.
4. Calculez le nombre d'opérations élémentaires du pire cas de l'algorithme validation_P.
5. Calculez la complexité de l'algorithme validation_P.

Solution :

1. Une instance I du problème de décision P consiste en deux entiers. Une structure de données permettant de coder (représenter) l'instance est un tableau de taille 2 d'entiers.
2. Un certificat c est tout simplement un entier pouvant être codé par une variable entière.
3. L'algorithme de validation validation_P est une fonction booléenne à deux paramètres : une instance I (tableau de taille deux d'entiers) ; et un certificat c (variable entière).

```
validation_P(I,c)
        début
        si c²=I[1] mod I[2] alors retourner VRAI sinon retourner FAUX finsi
        fin
```

4. Le nombre d'opérations du pire cas de l'algorithme est clairement 4 (une multiplication, une opération mod, une affectation, et une opération 'retourner')
5. La complexité de l'algorithme de validation est donc Θ(1) (complexité constante, donc polynomial).