

CSI 3505 - Examen Mi-Session

Date: Mardi, 30 octobre 2018
Durée: 1h20 (13:00-14:20), **Lieu:** LMX Pièce: 405

Professeur: N. Zaguia
Page 1 de 4

NOM:

D'ÉTUDIANT(E):

1. C'est un examen à livre fermé. Les calculatrices ne sont pas permises.
2. Écrivez vos réponses sur ce questionnaire, dans les espaces alloués à cet effet sous chaque question. Si nécessaire, vous pouvez utiliser l'endos de la page comme brouillon.

POUR LES QUESTIONS 1, 2 ET 3, ON NE VOUS DEMANDE PAS D'EXPLIQUER VOS RÉPONSES.

Question 1. [10 points] Faites une analyse dans le pire cas de chacun des algorithmes ci-dessous et écrivez **la complexité en terme de Θ** . Les multiplications sont les seules opérations élémentaires que vous devez compter. Dans les deux cas, on suppose que n représente la taille du problème.

a) [5 points]

```
resultat = 25;
for (i = 1; i <= n; i++) {
    for (k = 1; k <= i2; k++) {
        resultat = resultat * 10;
    }
}
```

Complexité :

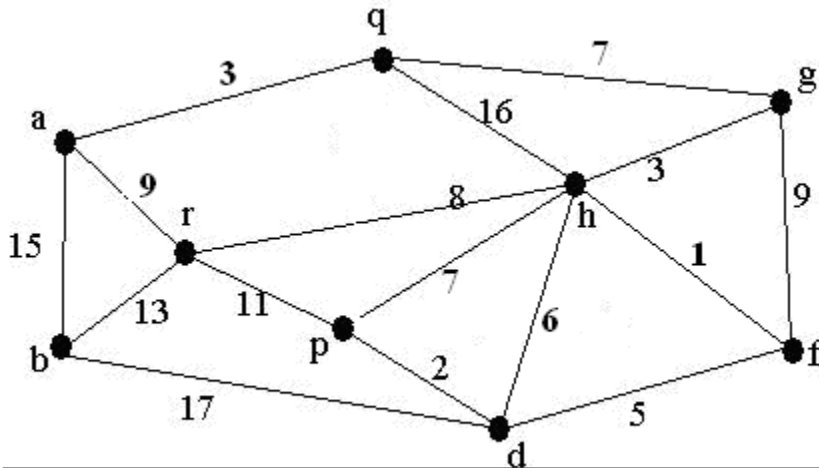
b) [5 points]

```
int Power1(int x, int n)
{
    if (n==1)
        return x;
    else
        return x * Power1(x, n-1); //*****
}
```

Complexité :

Question 2. [5 points]

On considère le graphe G ci-dessous. Dans quel ordre l'algorithme de Prim va ajouter les sommets de G à l'arbre recouvrant minimal ? On suppose que l'algorithme de Prim commence avec le sommet r.



Réponse :

Question 3. [5 points]

On considère l'instance ci-dessous du problème d'ordonnancement de tâches avec échéances. (n tâches à exécuter une à la fois- chacune en temps unitaire - la tâche i apporte un gain $g_i > 0$ si et seulement si elle est exécutée au plus tard à l'instant « échéance »)

<u>tâche</u>	<u>échéance</u>	<u>gain</u>
1	3	10
2	3	6
3	3	12
4	3	8

En utilisant l'algorithme vorace vu en classe, quel est l'ordonnancement final?

- A) tâche 1 au temps 1, tâche 2 au temps 2, tâche 3 au temps 3.
- B) tâche 3 au temps 1, tâche 4 au temps 2, tâche 1 au temps 3.
- C) tâche 3 au temps 1, tâche 1 au temps 2, tâche 4 au temps 3.
- D) tâche 4 au temps 1, tâche 1 au temps 2, tâche 3 au temps 3.
- E) tâche 3 au temps 1, tâche 2 au temps 2, tâche 1 au temps 3.

Question 4. [10 points]

a) [4 points] On considère l'algorithme *Resultat* ci-dessous :

```
int Resultat(int n)
{
  si (n<=2)
    retourne 3;
  sinon
    retourne 4*Resultat(n-1) + 6*Resultat(n-3) + 3 ;
}
```

Réponse a)

Soit $W(n)$ la complexité dans le pire cas de l'algorithme *Resultat*. On suppose que les seules opérations élémentaires qu'on compte sont les multiplications. Donner une relation récursive pour $W(n)$. (*On ne vous demande pas de donner des explications à votre réponse ni de résoudre l'équation.*)

b) [6 points] Écrivez un algorithme (en pseudocode) de type "**programmation dynamique**" qui calcule *Resultat*(n). Cet algorithme doit avoir une complexité de $\Theta(n)$

Question 5. [5 points]

Recherche d'un élément x dans une liste: l'algorithme de recherche séquentielle traverse le tableau à partir du premier élément jusqu'à ce que l'on trouve x , Sinon on arrive à la fin du tableau. Donner la complexité **dans le cas moyen** de l'algorithme de recherche séquentielle. On suppose que la probabilité que x soit dans la liste est q . **“expliquer en détails votre réponse”**