ROYAUME DU MAROC
Ministère de l'Enseignement Supérieur, de la Recherche Scientifique et de la Formation des Cadres

Université Hassan II Casablanca

Ecole Normale Supérieure de l'Enseignement Technique Mohammedia

المملكة المغربية

وزارةالتعليم العالي والبحث العلمي وتكوين الاطر

جامعة الحسن الثاني

الدار البيضاء

المدرسة العليا لاساتذة التعليم التقني

المحمدية

Concours d'accès en première année du cycle d'ingénieurs

Génie du Logiciel et des Systèmes Informatiques Distribués (*GLSID)*

Session : Septembre 2014

Epreuve d'Informatique

Durée : 3 heures

Remarques importantes :

- L'usage de la calculatrice ou de tout autre appareil électronique est interdit.
- Aucun document n'est autorisé.
- Les réponses à toutes les questions doivent être rédigées dans les feuilles de réponses.
- Chaque question du QCM ne peut avoir qu'une seule réponse possible parmi les quatre choix dont la lettre correspondante (A, B, C ou D) est à reporter dans les feuilles de réponses.
- Les solutions algorithmiques peuvent être rédigées en utilisant un pseudo langage algorithmique ou l'un des langages de programmation suivants : C, C++, C#, Java
- La clarté et la précision de votre solution algorithmique sera prise en considération.
- L'épreuve est notée sur un total de 100 points.

QCM : (10 points)

Reportez dans les feuilles de réponses, la lettre qui correspond à la bonne réponse pour les questions suivantes :

1. On insère les éléments 4, 3, 12, 7, 9 (dans cet ordre) dans une *pile*. Dans quel ordre vont-ils ressortir ?

   A) 9, 7, 12, 3, 4
   B) 3, 4, 7, 9, 12
   C) 4, 3, 12, 7, 9
   D) 12, 9, 7, 4, 3

2. On insère les éléments 4, 3, 12, 7, 9 (dans cet ordre) dans un *tas*. Dans quel ordre vont-ils ressortir ?

   A) 9, 7, 12, 3, 4
   B) 3, 4, 7, 9, 12
   C) 4, 3, 12, 7, 9
   D) 12, 9, 7, 4, 3

3. A laquelle des structures suivantes s'apparente le plus une représentation de graphe par listes de successeurs ?

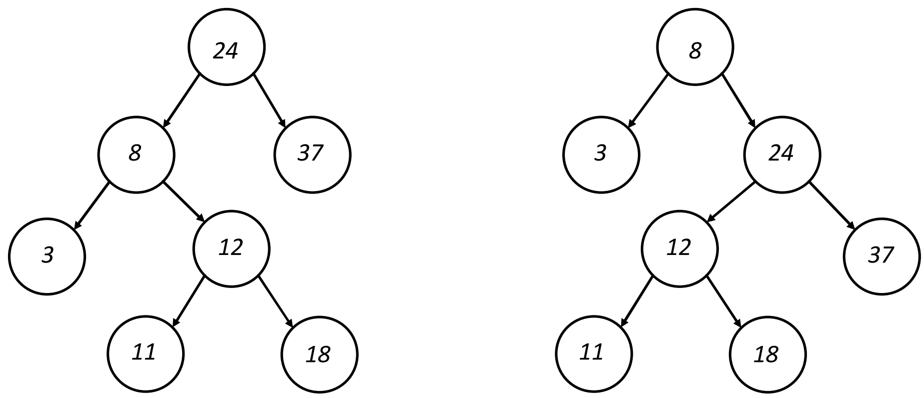
   A) une pile,
   B) un arbre binaire,
   C) une table de hachage,
   D) un tableau bidimensionnel.

4. Quelle est la complexité dans le pire cas de la recherche d'un élément dans un arbre binaire de recherche de hauteur h contenant n nœuds ?

   A) $\Theta(n)$
   B) Θ (log n)
   C) Θ (h)
   D) Θ (log h)

5. Quelle opération transforme l'arbre de gauche de la figure ci dessous en celui de droite ?

   A) une rotation droite,
   B) une double rotation,
   C) une rotation gauche,
   D) aucun des trois.



6. Comment calcule-t-on généralement la complexité d'un algorithme récursif ?

    A) On lance plusieurs fois l'algorithme avec différentes tailles de données,
    B) On établit puis on résout une formule de récurrence,
    C) On traduit l'algorithme en algorithme itératif et on regarde les boucles,
    D) On calcule des probabilités.

7. Le parcours en profondeur d'un arbre binaire correspond à un fonctionnement de :

    A) File (First In First Out) ,
    B) Pile (First In Last Out) ,
    C) Liste chaînée,
    D) Graphe orienté.

8. Parmi ces algorithmes de tri, lequel est un algorithme de type "Diviser pour régner" ?

    A) Le tri à bulles,
    B) Le tri par insertion,
    C) Le tri rapide (QuickSort),
    D) Le tri par tas.

9. On souhaite calculer tous les plus courts chemins d'un nœud donné à tous les autres nœuds dans un graphe orienté, qui peut contenir des cycles et dont les arcs peuvent avoir des poids négatifs, mais sans cycle absorbant. Quel est le meilleur algorithme pour résoudre ce problème ?

    A) L'algorithme qui fait un tri topologique des nœuds,
    B) Bellman-Ford,
    C) Dijkstra,
    D) Floyd-Warshall.

10. L'intérêt du tri par tas, comparativement aux autres algorithmes de tri, est :

    A) sa complexité en meilleur cas,
    B) sa complexité moyenne,
    C) sa complexité en pire cas,
    D) la place mémoire nécessaire.

Exercice 1 : (10 points)

1. Écrire un algorithme qui calcule le développement limité à l'ordre n de la fonction sin x définie par :

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \cdots$$

2. Evaluer la complexité de votre solution

Exercice 2 : (16 points)

On considère la suite définie par :

$$\begin{cases} U(0)=0 \\ U(1)=1 \\ U(n)=U(n-2)+U(n-1) \quad \text{pour } n > 1. \end{cases}$$

1. Ecrire un algorithme récursif qui calcule le $n^{\text{ième}}$ terme de cette suite
2. Donner l'arbre des appels récursifs pour calculer U(5)
3. Montrer par récurrence que la complexité de cette solution pour calculer U(n) est en ordre de $\left(\frac{1+\sqrt{5}}{2}\right)^n$
4. Quels sont les inconvénients de cette solution
5. Proposer une solution itérative permettant d'éviter les inconvénients de la solution récursive
6. Quel est l'ordre de la complexité de cette solution
7. Proposer une solution algorithmique permettant d'étudier la convergence du rapport U(n-1)/U(n)

Exercice 3 : (12 points)

Ecrire un algorithme qui permet de fusionner deux listes triées T1 et T2 de tailles respectives N1 et N2 dans une liste triée T.

Exemple :

Pour T1 = {1 , 7, 9, 15 } et T2= {5, 6, 8, 18, 22}, le résultat serait T={1,5,6,7,8,9,15,18,22}

Exercice 4 : (16 points)

Soit *T* un tableau de *n* entiers. On souhaite localiser les deux éléments distincts ayant les valeurs les plus proches. Autrement dit, les deux éléments dont la différence en valeur absolue est la plus petite.

Exemple :

0	1	2	3	4	5	6	7	8	9
100	30	100	17	11	12	7	18	200	25

Pour cet exemple on cherche à produire les résultats suivants :

- Ecart entre les deux éléments est : 18-17=1
- Positions des éléments sont : 3, 7

1. Ecrire, dans le cas d'un tableau non trié, un algorithme optimal qui recherche les deux éléments les plus proches dans *T*. l'algorithme doit retourner les positions de ces deux éléments ainsi que leur écart.
2. Donner l'ordre de grandeur de la complexité de votre solution.
3. On considère une méthode de tri nommée ***Tri_Rapide*** capable de trier un tableau *T* de *n* éléments avec une complexité moyenne de l'ordre de *n log(n)*. Donner un deuxième algorithme qui commence forcement par l'appel à la méthode ***Tri_Rapide*** pour trier d'abord le tableau *T* avant de commencer la recherche des deux éléments les plus proches.
4. Quel est l'ordre de grandeur de la complexité de votre deuxième solution. Conclure

Exercice 5 : (18 points)

On considère une image monochrome (256 niveaux de gris) stockée dans une matrice de taille *N* x *M* où *N* représente le nombre de lignes et *M* le nombre de colonnes. Chaque élément de cette matrice représente le niveau de gris d'un pixel de l'image dont la valeur est comprise entre 0 et 255.

En imagerie numérique, l'histogramme représente la distribution des intensités (ou des couleurs) de l'image. Pour une image monochrome, l'histogramme est défini comme une fonction discrète qui associe à chaque niveau de gris le nombre de pixels de l'image ayant cette valeur. La détermination de l'histogramme est donc réalisée en comptant le nombre de pixels pour chaque niveau de gris.

Dans le domaine du traitement d'images, l'opérateur ***Sobel*** est utilisé particulièrement avec les algorithmes de détection du contour. C'est un opérateur de différentiation discrète calculant une approximation du gradient de la fonction d'intensité de l'image. Il est basé sur le calcul du produit de convolution de l'image avec un filtre dans les deux directions verticale et horizontale. L'opérateur utilise deux matrices de convolution *H* et *V* de taille 3 x 3 qui seront appliquées à l'image originale pour calculer respectivement les deux approximations des dérivées notées *Gx* et *Gy* où *Gx* représente les changements horizontaux et *Gy* les changements verticaux. Si on définit *A* comme l'image source, les approximations *Gx* et *Gy* sont données par :

$$Gx = H * A \;\; et \;\; Gy = V * A$$

$$Avec \;\; H = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \;\; et \;\; V = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Où * représente le produit de convolution à deux dimensions exprimé, à titre d'exemple pour le cas de *Gx* par :

$$Gx\,(i,j) = \sum_{k=-1}^{1} \sum_{l=-1}^{1} H(k,l)A(i+k, j+l)$$

Avec i et j représentent respectivement le numéro de la ligne et le numéro de la colonne du pixel de l'image.

Dans chaque point de l'image, le gradient résultat *G* peux être la combinaison des deux gradients *Gx* et *Gy*.

$$\mathrm{G} = \sqrt{\mathrm{Gx}^2 + \mathrm{Gy}^2}$$

1. Ecrire un algorithme qui permet de stocker, ligne par ligne, une image monochrome représentée par une matrice *N* lignes et *M* colonnes, dans un vecteur de taille *N* x *M*.
2. Ecrire un algorithme qui permet de calculer l'histogramme d'une image monochrome stockée dans un vecteur de taille *N* x *M*.
3. Ecrire l'algorithme qui permet de déterminer le contour *G* d'une image monochrome stockée dans un vecteur de taille *N* x *M*, en appliquant l'opérateur ***Sobel*** décrit ci-dessus.

Exercice 6 (18 points)

On considère un polynôme de degré n à coefficients a_i et à variable x réels donné par le schéma usuel suivant :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

Pour représenter ce polynôme on peut utiliser, parmi les solutions, une liste doublement chaînée dont les nœuds représentent les termes du polynôme. Chaque terme est un monôme défini par son coefficient et son degré.

Exemple :

Le polynôme $P(x) = 4x^7 - 7x^3 + 6$

Est représenté par la liste P= {(4,7) ;(-7,3) ;(6,0)}

1. Ecrire la déclaration de la structure qui représente un monôme.
2. Ecrire la déclaration de la structure qui représente un polynôme sous forme d'une liste doublement chaînée.
3. Ecrire la fonction qui permet de calculer la somme de deux polynômes P et Q.
4. Ecrire la fonction qui permet de calculer le produit deux polynômes P et Q.
5. Ecrire la fonction qui permet de trouver les solutions réelles dans un intervalle [a, b], si elles existent, de l'équation P(x)=0 où P est un polynôme donné.