UNIVERSITE SAAD DAHLAB – BLIDA 1
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATQUE

CONCOURS D'ACCES A LA FORMATION DE TROISIEME CYCLE EN VUE DE L'OBTENTION DU DOCTORAT EN INFORMATIQUE

AU TTIRE DE L'ANNEE UNIVERSITAIRE 2019/2020

EPREUVE D'ALGORITHMIQUE AVANCEE

Exercice 1 (10 pts)

- *Dans les questions à choix multiple, cocher la ou les bonnes réponses.*
- *Dans une question à réponses multiples, une réponse fausse élimine une réponse juste.*

1. Comment calcule-t-on 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.

2. Un algorithme déterministe est un algorithme dont la solution est :
   A. Déduite à partir de ses spécifications.
   B. Devinée puis vérifiée.
   C. Introuvable
   D. Difficilement trouvable

3. Un algorithme non déterministe est un algorithme dont la solution est :
   A. Déduite à partir de ses spécifications.
   B. Devinée puis vérifiée.
   C. Introuvable
   D. Difficilement trouvable

4. La classe P regroupe tous les problèmes de décision qui peuvent être résolus par :
   A. un algorithme déterministe de complexité polynomiale
   B. un algorithme déterministe de complexité exponentielle
   C. un algorithme non déterministe de complexité polynomiale
   D. un algorithme non déterministe de complexité exponentielle

5. La classe NP regroupe tous les problèmes de décision qui peuvent être résolus par :
   A. un algorithme déterministe de complexité polynomiale
   B. un algorithme déterministe de complexité exponentielle
   C. un algorithme non déterministe de complexité polynomiale
   D. un algorithme non déterministe de complexité exponentielle

6. Soient A et B deux problèmes de décision de classe NP, et supposons qu'on connaisse une transformation polynomiale (une réduction) de A en B. Lesquelles de ces propositions sont vraies ?
   A. A est plus difficile que B.
   B. Si $A \in P$ alors $B \in P$.
   C. Si $A \in$ NP-complet alors $B \in$ NP-complet.
   D. Si on connaît une transformation polynomiale de B en A alors A et B sont équivalents.
   E. B est plus difficile que A.

F. Si $B \in P$ alors $A \in P$.

G. Si $B \in$NP-complet alors $A \in$NP-complet.

H. Si A et B sont NP-Complet alors il existe une transformation polynomiale de B en A

7. Soient les classes de complexité suivantes : P, NP, NP-complet et NP-difficile, lesquelles de ces propositions sont vraies ?

A. $P \subset NP$

B. $NP \subset$ NP-Complet

C. NP-Complet $\cap$ NP-Difficile $= \varnothing$

D. $P \cap$ NP- Complet $= \varnothing$

E. $P \subset NP \subset$ NP-Complet $\subset$ NP-Difficile

F. $NP \subset P$

G. NP-Complet $\subset NP$

H. NP-Complet $\cap$ NP-Difficile $\neq \varnothing$

I. $P \cap$ NP- Complet $\neq \varnothing$

J. NP- Difficile $\subset$ NP-Complet $\subset NP \subset P$

8. Dans un problème K-SAT, toutes les clauses contiennent

A. au moins k littéraux

B. au plus k littéraux

C. exactement k littéraux

9. Dans un problème MAX-SAT, il s'agit de trouver une affectation des variables booléennes qui

A. minimise le nombre des clauses à faux.

B. maximise le nombre des clauses à faux.

C. évalue toutes les clauses à vrai.

D. évalue toutes les clauses à faux.

10. Soit la clause $C = P \vee Q$, Elle est équivalente à

A. $(P \vee Q \vee Y1) \wedge (\neg Y1 \vee P \vee Q)$

B. $P \vee Q \vee T$

C. $(P \vee \neg Y1) \wedge (Y1 \vee Q)$

D. Q

11. Pour un problème 3-SAT contenant n variables booléennes et m clauses, on peut construire un graphe 3-coloriable qui contient :

A. « 3 + 2 n + 5 m » sommets.

B. « 2 m + 1 » triangles.

C. « m » motifs 3-coloriable.

D. « 2 n + 5 m » sommets.

E. « 2 n + 1 » triangles.

F. « n » motifs 3-coloriable.

12. Trouver la bonne description de chaque paradigme de programmation

A. Programmation dynamique

B. Diviser pour régner

C. Récursivité

D. Approche gloutonne

a. Partitionner le problème en sous-problèmes indépendants qui sont résolus de manière récursive, puis combiner leurs solutions pour résoudre le problème initial.

b. Définir un algorithme en fonction de lui-même.

c. Construire la solution optimale pas à pas.

d. Partitionner le problème en sous-problèmes qui ne sont pas forcément indépendants, puis combiner leurs solutions pour résoudre le problème initial.

13. Trouver le principe de chaque algorithme de tri

A. Tri par Sélection

B. Tri à bulles

C. Tri par Insertion

a. Permuter tous les éléments de telle sorte que tous ceux qui sont inférieurs au pivot soient à sa gauche et que tous ceux qui sont supérieurs au pivot soient à sa droite.

b. Insérer un élément dans une liste d'éléments déjà triés.

c. Inverser deux éléments successifs s'ils ne sont pas classés dans le bon ordre et de recommencer jusqu'à ce qu'on ne peut plus inverser.

D. Tri par TAS

E. Tri Rapide

F. Tri par ABR

G. Tri par Fusion

d. Transformer le tableau en un d'un arbre binaire de recherche, ensuite faire un parcours infixe de l'arbre.

e. Placer le plus petit élément au début du tableau, puis le second plus petit élément du tableau à la deuxième position et ainsi de suite jusqu'à ce que le tableau soit entièrement trié.

f. Fusionner deux tableaux triés de sorte à ce que le tableau final soit trié.

g. Transformer le tableau en un arbre binaire parfait dont le minimum se trouve à la racine, ensuite extraire la racine jusqu'à ce que l'arbre soit vide.

14. Parmi ces algorithmes de tri, lesquels utilisent le paradigme "Diviser pour Régner" ?

A. Bulles
B. Insertion
C. Rapide
D. Sélection
E. Fusion
F. TAS
G. ABR
H. Aucun

15. Une heuristique

A. est une règle empirique basée sur l'expérience.
B. est une règle analytique basée sur des concepts théoriques.
C. tient en compte de la spécificité du problème traité.
D. est décrite de manière abstraite, sans faire appel à un problème spécifique.

16. Trouver la bonne description de chaque méthode pour résoudre un problème d'optimisation

A. Méthode exacte

B. Méthode méta-heuristique

a. permet de trouver toujours la solution optimale.
b. permet de trouver une bonne solution (presque optimale).
c. nécessite un temps d'exécution raisonnable.
d. nécessite un temps d'exécution prohibitif.
e. se base sur une exploration complète de l'espace de solution.
f. se base sur une exploration partielle et guidée de l'espace de solution.

EXERCICE 2 (4 + 6= 10 pts) : *Les parties sont indépendantes*

Partie I :

1. Expliquer la propriété d'ordre (la relation d'ordre entre la valeur du père et ses fils) et la propriété structurelle de chacun de ces arbres : AVL, TAS_{min} et B-arbre d'ordre m.

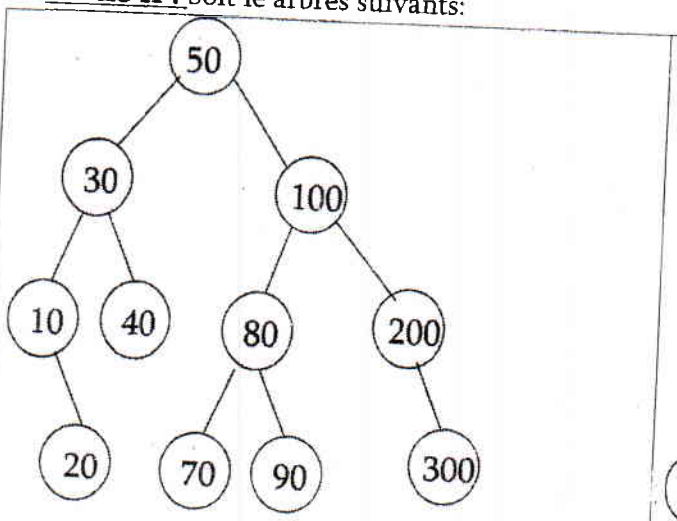
Arbres	Propriété d'ordre	Propriété structurelle
AVL		
TAS_{min}		
B-arbre d'ordre m		

2. Quelle est la complexité temporelle des opérations d'insertion et de suppression de chacun de ces arbres : AVL, TAS_{min} et B-arbre ? Choisir une bonne réponse parmi ces propositions (« n » présente le nombre des nœuds et « m » présente l'ordre de l'arbre) :

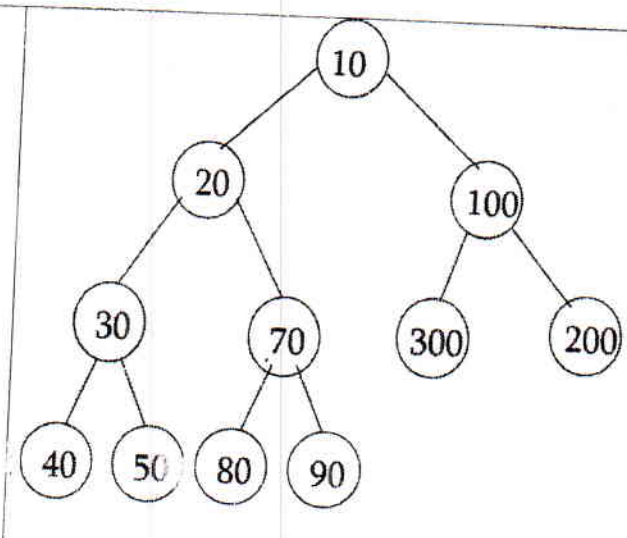
A. O (n)
B. O (m)
C. O (n *m)
D. O (n + m)
E. O ($log_m n$)
F. O ($log_n m$)

Arbres	Opération d'Insertion	Opération de suppression
AVL		
TAS_{min}		
B-arbre		

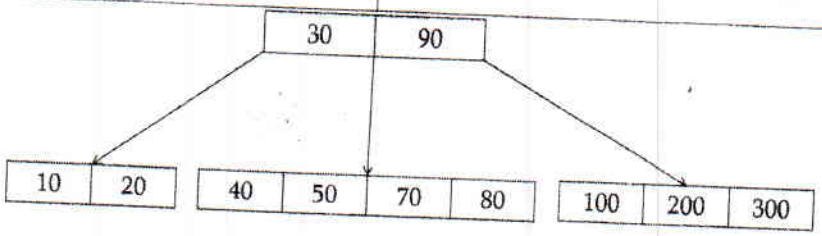
Partie II : Soit le arbres suivants:



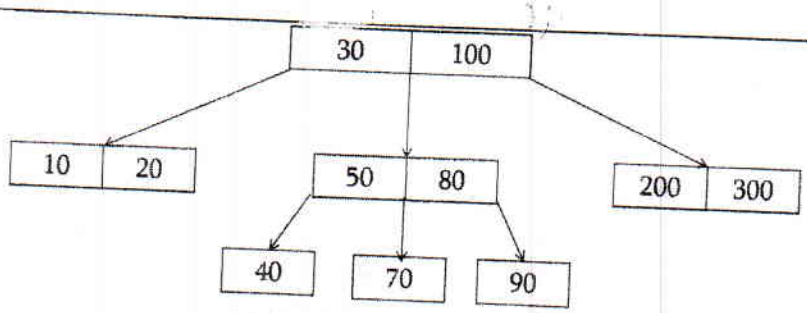
ARBRE 1



ARBRE 2



ARBRE 3



ARBRE 4

1. Spécifier le type de chacun de ces arbres. Choisir une bonne réponse parmi ces propositions :
   A. ABR (Arbre Binaire de Recherche)
   B. AMR (Arbre M-aire de Recherche)
   C. AVL
   D. TAS
   E. B-arbre
2. Donner l'ordre de chacun de ces arbres.
3. Insérer la valeur « 60 » dans chacun de ces arbres tout en préservant leur type (Montrer toutes les étapes nécessaires).
4. Supprimer la valeur « 30 » de chacun des arbres obtenus de la question précédente, tout en préservant leur type (Montrer toutes les étapes nécessaires).