

NOM :

Prénom :

- **aucun document n'est autorisé.**
- ce QCM aboutit à une note sur 50 points. La note finale sur 20 sera obtenue simplement en divisant la note sur 50 par 2,5.
- n'oubliez pas de remplir votre nom et votre prénom juste au dessus de ce cadre.

Chaque bonne réponse rapporte 1 point. Chaque mauvaise réponse enlève 1 point. Il n'y a qu'une seule bonne réponse par question. Ne répondez pas au hasard, la note totale peut être négative!

1] Laquelle de ces complexités est la plus petite ?

- $\Theta(n^n)$ $\Theta(n!)$ $\Theta(2^{2^n})$ $\Theta(2^n)$

2] Lequel de ces algorithmes de tri a la moins bonne complexité en moyenne ?

- tri fusion, tri rapide, tri par insertion, tri par tas.

3] Lequel de ces algorithmes de tri a la meilleure complexité dans le pire cas ?

- tri fusion, tri rapide, tri par insertion, tri à bulles.

4] Lequel de ces algorithmes de tri ne permet pas de trier un tableau *en place* (sans réallouer de mémoire) ?

- tri fusion, tri rapide, tri par insertion, tri à bulles.

5] Quelle est la complexité (en moyenne) minimale d'un algorithme de tri par comparaisons de n éléments ?

- $\Theta(\log n)$, $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n^2)$.

6] Laquelle de ces étapes ne fait pas partie de la stratégie « diviser pour régner » ?

- diviser le problème, trier les sous-problèmes,
 résoudre les sous-problèmes, recombinaison des résultats.

7] Sachant qu'il faut déplacer 7 disques pour résoudre le problème des tours de Hanoï à 3 disques, combien faut-il en déplacer (au minimum) pour résoudre le problème comportant 4 disques ?

- 8, 14, 15, 49.

8] On veut appliquer un algorithme de tri rapides aux entiers : $\{1, 4, 8, 2, 9, 7, 3, 6, 5\}$. Pour la première étape du tri, quel serait le meilleur choix de pivot ?

- 1, 9,
 5, cela n'a pas d'importance.

9] En calculant la complexité d'un algorithme récursif, on aboutit à la formule $T(n) = T(\frac{n}{2}) + 2$ et $T(0) = 1$. Quelle est la complexité $T(n)$ de cet algorithme ?

- $\Theta(\log n)$, $\Theta(n)$, $\Theta(n^2)$, $\Theta(2^n)$.

10] Les lignes suivantes affichent les éléments de quelle type de structure ?

```
1 while (a != NULL) {
2   cout << a->val << endl;
3   a = a->next;
4 }
```

- un tableau, une liste, un tas, un arbres binaire.

11] Laquelle de ces structures ne peut pas être programmée efficacement avec un tableau ?

- une pile, un tas, une file, une liste.

12] Quelle est la complexité de la désallocation (libération de la mémoire) d'une liste de n éléments ?

- $\Theta(1)$, $\Theta(\log n)$, $\Theta(n)$, $\Theta(n^2)$.

13] Laquelle de ces opérations ne s'effectue pas en temps constant sur une liste chaînée de n éléments ?

- insertion d'un élément en première position,
 suppression de l'élément en première position,
 accès à l'élément à la position $\frac{n}{2}$,
 insertion d'un élément à la position courante.

14] Quel devrait être environ la taille d'une table à adressage direct des définitions de tous les mots de 9 lettres ?

- 26×9 , le nombre de mots français de 9 lettres,
 9^{26} , 26^9 .

15] Laquelle de ces implémentations d'un dictionnaire propose la recherche la plus lente ?

- table à adressage direct, table pour recherche dichotomique,
 table pour recherche séquentielle, table de hachage.

16] Laquelle de ces opérations s'effectue toujours en temps constant dans une table de hachage ?

- insertion, suppression, recherche, aucune des trois.

17] On veut insérer l'élément $x=(\text{clef}, \text{info})$ dans une table de hachage tab dont la fonction de hachage est h . La fonction $\text{insert}(v, L)$ insère l'élément v dans la liste L . Laquelle de ces opérations est la bonne ?

- $\text{insert}(\text{clef}, \text{tab}[\text{info}])$, $\text{insert}(\text{tab}[\text{clef}], h(\text{info}))$,
 $\text{insert}(h(x), \text{tab}[\text{clef}])$, $\text{insert}(x, \text{tab}[h(\text{clef})])$.

18] On insère n éléments dans une table de hachage de taille k . Quel est son facteur de remplissage ρ ?

- $\rho = \frac{n}{k}$, $\rho = n - k$, $\rho = k - n$, $\rho = n \times k$.

19] Laquelle de ces affirmations concernant des arbres est *fausse* ?

- la racine peut être une feuille,
 un nœud interne possède toujours au moins un fils,
 la racine possède toujours au moins un fils,
 la racine ne possède jamais de père.

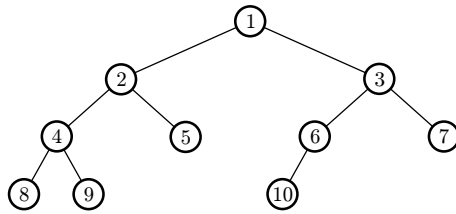


FIG. 1 – Un arbre binaire.

- 20] Dans l'arbre de la FIG. 1, le nœud 5 est :
- une feuille, un nœud interne, une racine, aucun des trois.
- 21] Un parcours de l'arbre de la FIG. 1 qui traite les nœuds dans l'ordre 1, 2, 3, 4, 5, 6, 7, 8, 9 puis 10 est un parcours :
- préfixe, infixe, postfixe, en largeur.
- 22] Un parcours de l'arbre de la FIG. 1 qui traite les nœuds dans l'ordre 8, 4, 9, 2, 5, 1, 10, 6, 3 puis 7 est un parcours :
- préfixe, infixe, postfixe, en largeur.
- 23] Un parcours de l'arbre de la FIG. 1 qui traite les nœuds dans l'ordre 8, 9, 4, 5, 2, 10, 6, 7, 3 puis 1 est un parcours :
- préfixe, infixe, postfixe, en largeur.
- 24] Dans un arbre binaire de recherche, le nœud contenant la plus grande clef :
- est la racine, est un nœud interne,
 est une feuille, peut être les trois.
- 25] Laquelle de ces trois affirmations est *fausse* ?
- un tas est un arbre complet,
 un tas est un arbre équilibré,
 un tas est un arbre binaire de recherche,
 les trois sont justes.
- 26] Dans un tas, la clef du fils droit d'un nœud X est toujours
- plus petite que la clef du fils gauche de X ,
 plus grande que la clef du fils gauche de X ,
 plus petite que la clef de X ,
 aucun de trois.
- 27] Pour insérer un élément dans un tas :
- on l'insère à la racine et on le fait descendre à sa place,
 on l'insère à la fin du tas et on le fait remonter à sa place,
 on cherche sa place et on l'y insère en déplaçant le nœud présent,
 on cherche un feuille plus grande que lui, et on l'insère comme son fils gauche.
- 28] Pour une implémentation standard de tas avec un tableau, lequel des tableaux suivants ne représente pas un tas bien ordonné ?
- | | | | | | | |
|----|----|----|----|----|----|----|
| 55 | 24 | 18 | 14 | 11 | 16 | 10 |
|----|----|----|----|----|----|----|

87	46	86	34	30	50	11
----	----	----	----	----	----	----
- | | | | | | | |
|----|----|----|----|----|----|--|
| 66 | 17 | 50 | 16 | 10 | 15 | |
|----|----|----|----|----|----|--|

37	23	29	17	25	11	
----	----	----	----	----	----	--

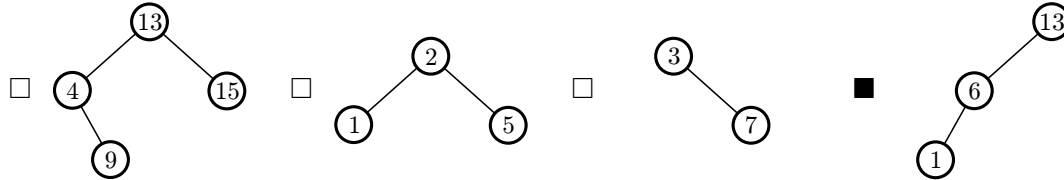
29] Quelle propriété d'un arbre *n'est pas toujours* conservée par une rotation ?

- son nombre de feuilles,
- son nombre de racines,
- son nombre de nœuds,
- son nombre d'arêtes.

30] Lors d'une double rotation pour le rééquilibrage d'un arbre AVL, deux nœuds sont utilisés comme départs pour une rotation dans un sens puis dans l'autre. Ces deux nœuds sont :

- un nœud puis son fils,
- un nœud puis son père,
- un nœud puis son frère,
- deux nœuds quelconques.

31] Parmi les arbres binaires de recherche suivants, lequel n'est pas un arbre AVL ?



32] Parmi ces quatre types d'arbres, lequel ne désigne pas un arbre binaire équilibré ?

- le tas,
- l'arbre 2-3,
- l'arbre rouge/noir,
- l'arbre AVL.

33] On considère un graphe orienté possédant a arcs et s sommets. Quel espace mémoire occupe sa représentation par liste de successeurs ?

- $\Theta(a)$
- $\Theta(s^2)$
- $\Theta(a + s)$
- $\Theta(a \times s)$

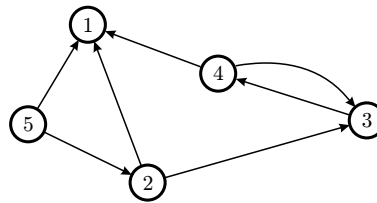


FIG. 2 – Un graphe orienté.

34] Quelle est la matrice d'adjacence du graphe de la FIG. 2 ?

- $\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
- $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$
- $\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$
- $\begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

35] Quelle est la longueur du *plus court* cycle du graphe de la FIG. 2 ?

- 2,
- 4,
- 5,
- il est sans cycle.

36] Parmi les quatre tableaux de pères suivants, lequel correspond à un recouvrement du graphe de la FIG. 2 par un arbre ?

- [- 1 | 4 | 1 | 1 |]
- [2 | 5 | 4 | 3 | -]
- [4 | 5 | 2 | 3 | -]
- [2 | - | 2 | 3 | 2]

37] Lors d'un parcours en largeur de graphe, on colorie les sommets en trois couleurs : blanc pour les sommets non visités, gris pour ceux en cours de visite et noir pour ceux déjà visités. De quelle couleur sont les sommets qui sont dans la file utilisée pour le parcours en largeur ?

- blancs, gris, noirs, gris ou noirs.

38] Qu'est-ce que *ne permet pas* de faire un parcours en profondeur d'un graphe ?

- détecter des cycles,
 construire un recouvrement du graphe,
 trouver un plus court chemin entre deux sommets,
 faire du tri topologique.

39] Le tri topologique d'un graphe permet :

- de remettre de l'ordre dans un graphe,
 d'effectuer des tâches interdépendantes dans le bon ordre,
 de passer d'une représentation de graphe à une autre,
 de trouver le nombre de chemins possibles entre deux sommets.

40] Quel est le nombre maximum d'arcs que peut posséder un graphe orienté à 8 sommets ?

- 16, 56, 64, 256.

41] L'algorithme de Aho, Hopcroft, Ullman permet de calculer beaucoup de choses dans un graphe : sa fermeture transitive, l'ensemble de ses plus courts chemins... Quelle est sa complexité pour un graphe à n sommets ?

- $\Theta(n \log n)$, $\Theta(n^2)$, $\Theta(n^3)$, $\Theta(n^4)$.

42] En utilisant l'algorithme naïf qui teste la présence du motif à chaque position du texte, quelle est la complexité (dans le pire cas) de la recherche d'un motif de taille m dans un texte de taille n ?

- $\Theta(m)$, $\Theta(n)$, $\Theta(n \times m)$, $\Theta(n^m)$,

43] En utilisant un automate fini, quelle est la complexité (dans le pire cas) de la recherche d'un motif de taille m dans un texte de taille n (sans compter le coût de construction de l'automate) ?

- $\Theta(m)$, $\Theta(n)$, $\Theta(n \times m)$, $\Theta(n^m)$,

44] Combien d'états comporte un automate fini utilisé pour reconnaître un motif de taille m dans un texte de taille n ?

- m , $m + 1$, n , $m + n$.

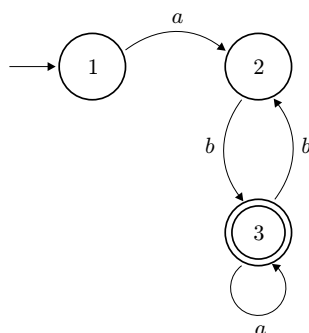


FIG. 3 – Un automate fini.

45] Quel est le langage reconnu par l'automate de la FIG. 3 ?

- $ab(bb)^*a^*$, $a(bab)^*b$, $ab(a|bb)^*$, $a(bab|bb)^*ba^*$.

46] Laquelle de ces structures de données utiliseriez-vous pour implémenter efficacement une file de priorité ?

- une liste,
- un tas,
- une table de hachage,
- un graphe.

47] On veut ranger n entiers dans une structure de donnée de façon à pouvoir ajouter de nouveaux entiers et en supprimer en temps $\Theta(\log n)$. On veut aussi pouvoir chercher le plus petit entier plus grand qu'un entier donné en temps $\Theta(\log n)$ dans le pire cas. Quelle structure proposez-vous d'utiliser ?

- une liste,
- un tas,
- une table de hachage,
- un arbre AVL.

48] On suppose que l'on a accès en temps constant à une mémoire de taille infinie. Quelle est alors la structure la plus efficace pour l'implémentation d'un dictionnaire ?

- une table à adressage direct,
- une table de hachage,
- un arbre binaire de recherche équilibré,
- une liste de mots.

49] L'algorithme de Rabin-Karp est utilisé pour :

- l'équilibrage d'arbres,
- la recherche de motif,
- le parcours de graphe,
- l'allocation dynamique de mémoire.

50] Laquelle de ces opérations a la complexité dans le pire cas la plus petite (en utilisant le meilleur algorithme possible) ?

- trier un tableau de n entiers,
- rechercher un élément et le supprimer dans une liste de n entiers,
- trouver tous les successeurs d'un sommet d'un graphe à n sommets,
- calculer la puissance n -ième d'un entier modulo 1037.