

NOM :

Prénom :

- **aucun document n'est autorisé.**
- ce QCM aboutit à une note sur 42 points. La note finale sur 20 sera obtenue simplement en divisant la note sur 42 par 2. Il suffit donc de donner 20 réponses justes (et aucune fausse) pour avoir la moyenne.
- 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. Vous n'êtes pas obligés de répondre à toutes les questions : une question sans réponse compte pour 0. Ne répondez donc pas au hasard !

1] Quelle ligne du code suivant contient une erreur de syntaxe qui empêche la compilation ?

```
1 void f(unsigned int n) {
2   int i; j = 1;
3   for (i=1; i<n; i++) {
4     j = i*i*j;
5   }
6   printf("Resultat : %d -> %d\n", n, j);
7 }
```

- 1 2 4 6

2] La fonction main d'un programme écrit en C prend 2 arguments. Quel est le type du premier de ces arguments ?

- int* *char** *char*** cela dépend.

3] Laquelle des fonctions suivantes retourne le nombre de lettres de la chaîne de caractères passée en argument ?

```
1 int strlen(char* m) {
2   int i;
3   while (m[i] == 0) {
 4     i = i + 1;
5   }
6   return i;
7 }
```

```
1 int strlen(char* m) {
2   int i = 1;
3   while (i > 0) {
 4     if (m[i] = 0) { i = 0;}
5   }
6   return m[i];
7 }
```

```
1 int strlen(char* m) {
2   int i = 1;
3   while (i > 0) {
 4     i = m[i];
5   }
6   return i;
7 }
```

```
1 int strlen(char* m) {
2   int i = 0;
3   while (m[i] != 0) {
 4     i++;
5   }
6   return i;
7 }
```

- 4] En C, un `int` est un entier signé, codé sur 32 bits, dont le bit de poids fort sert à déterminer le signe : 0 pour un nombre positif, 1 pour un nombre négatif. Quelle est la valeur maximale que peut prendre un `int` ?
- $2^{15} - 1$ $2^{16} - 1$ $2^{31} - 1$ $2^{32} - 1$
- 5] Laquelle de ces valeurs est la plus grande ?
- `sizeof(int*)` `sizeof(char*)`
 `sizeof(double*)` les 3 ont la même taille.
- 6] Dans une boucle `for(A;B;C)` où A, B et C représentent des instructions, quelles sont les instructions qui seront exécutées en premier (avant le début de la boucle) et en dernier (à la fin de la boucle) ?
- A en premier, C en dernier B en premier, B en dernier
 A en premier, B en dernier B en premier, C en dernier
- 7] Que mesure la complexité spatiale d'un algorithme ?
- Le nombre de lignes de code nécessaires pour l'écrire.
 La taille du fichier exécutable généré par le compilateur.
 La quantité de mémoire nécessaire à son exécution.
 Le nombre d'appels à la fonction `malloc`.
- 8] Quelle est la complexité du meilleur algorithme possible pour calculer la somme des carrés des entiers de 1 à n , $\sum_{i=1}^n i^2$? (on considère que n est suffisamment petit pour que les additions et les produits se fassent en temps constant)
- $\Theta(1)$ $\Theta(\log n)$ $\Theta(n)$ $\Theta(n^2)$
- 9] Quelle est la complexité d'un algorithme naïf effectuant le produit de deux matrices carrées de taille $n \times n$?
- $\Theta(n^2)$ $\Theta(n^3)$ $\Theta(n^4)$ $\Theta(n \log n)$
- 10] On veut trier un tableau de n entiers à l'aide d'un algorithme de tri à bulles standard (chaque itération déplace le plus grand élément à la fin du tableau). Combien d'itérations (au minimum) faut-il effectuer pour être certain que le tableau soit entièrement trié ?
- 1 $n - 1$ n n^2
- 11] Lors du *tri fusion* d'un tableau de n éléments, combien de fois au total la fonction `merge` qui fusionne les deux sous-tableaux triés est-elle appelée ?
- 1 fois $\log n$ fois $\frac{n}{2}$ fois $n - 1$ fois
- 12] Lequel des algorithmes de tri suivants est le plus rapide (asymptotiquement) si on lui donne en entrée un tableau inversement trié ?
- tri par insertion tri fusion tri à bulles tri cocktail
- 13] Quelle est la complexité en moyenne minimale d'un algorithme de tri par comparaisons appliqué à des tableaux de taille n uniformément distribués ?
- $\Theta(\log n)$ $\Theta(n)$ $\Theta(n \log n)$ $\Theta(n^2)$
- 14] Complétez la phrase : l'algorithme de résolution du problème des tours de Hanoï est un bon exemple...
- ...d'algorithme récursif. ...d'algorithme par force brute.
 ...d'algorithme glouton. ...d'algorithme polynomial.

15] En plus d'un (ou plusieurs) appels à lui-même, que doit toujours comporter un algorithme récursif?

- une boucle `for` ou `while`
- une étape de division
- une condition d'arrêt
- une étape de fusion

16] Lors de la compilation d'un programme, l'*inline* consiste à remplacer un appel de fonction par une copie du code de la fonction appelée. Quel type de fonction ne peut pas être compilée en *inline*?

- les fonctions sans arguments,
- les fonctions sans valeur de retour,
- les fonctions récursives,
- les fonctions utilisant plusieurs boucles imbriquées.

17] Dans une implémentation standard de liste chaînée (comme vue en cours), comment peut-on reconnaître le dernier élément de la liste?

- il contient la valeur -1 ,
- c'est le plus grand élément de la liste,
- il pointe sur lui-même,
- il contient un pointeur vers NULL.

18] Un arbre binaire contient 4 nœuds internes (dont sa racine) et 5 feuilles. Quel est son nombre total de nœuds?

- 8
- 10
- 9
- ce n'est pas possible.

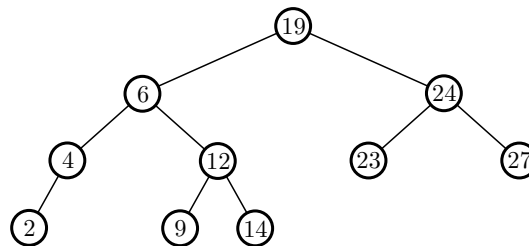


FIGURE 1 – Un arbre binaire de recherche.

19] Si l'on effectue un parcours en largeur de l'arbre de la FIGURE 1, dans quel ordre ses nœuds s'afficheront-ils?

- 2, 4, 6, 9, 12, 14, 19, 23, 24, 27
- 19, 6, 4, 2, 12, 9, 14, 24, 23, 27
- 19, 6, 24, 4, 12, 23, 27, 2, 9, 14
- 2, 4, 9, 14, 12, 6, 23, 27, 24, 19

20] Si l'on insère la valeur 5 dans l'arbre binaire de recherche de la FIGURE 1, quel nœud sera le père du nouveau nœud inséré?

- 4
- 6
- 9
- cela dépend

21] Considérons maintenant l'arbre de la FIGURE 1 comme un arbre AVL. La suppression duquel des nœuds suivants obligerait à faire une rotation pour rééquilibrer l'arbre?

- 4
- 23
- 9
- il restera équilibré dans tous les cas.

22] Considérons encore l'arbre de la FIGURE 1 comme un arbre AVL. L'insertion de l'élément 11 crée un déséquilibre. Quel nœud sera à la racine de l'arbre après rééquilibrage?

- 6
- 12
- 19
- 24

23] Soit h une fonction de hachage de $[0, m - 1]$ dans $[0, k - 1]$. On construit une table de hachage à l'aide de h et on y insère n éléments différents. Quelle est alors la complexité (en moyenne) de l'insertion d'un $(n + 1)$ -ème élément ?

- $\Theta(1)$
 $\Theta(\frac{n}{k})$
 $\Theta(\frac{m}{n})$
 $\Theta(\log n)$

24] Laquelle des propriétés suivantes est nécessaire et suffisante pour que la relation d'ordre dans un tas soit respectée ?

- Le plus grand élément de chaque sous-arbre du tas est à la racine de ce sous-arbre.
 Les valeurs des nœuds d'un niveau du tas sont supérieures à toutes celles des nœuds du niveau d'en dessous.
 Le plus grand élément du tas est sa racine.
 La valeur du fils gauche d'un nœud est inférieure à celle du fils droit.

25] Dans un tas de capacité maximale m contenant n éléments, quelle est *dans le pire cas* la complexité de l'extraction (recherche et suppression) de l'élément de priorité maximale ?

- $\Theta(m)$
 $\Theta(n \log n)$
 $\Theta(m \log m)$
 $\Theta(\log n)$

26] L'algorithme de Dijkstra permet la recherche de plus courts chemins dans un graphe pondéré, orienté ou non. Pour qu'il fonctionne, le graphe doit avoir l'une des propriétés suivantes, laquelle ?

- Le graphe doit être sans cycles.
 Les poids des arcs/arêtes doivent être positifs.
 Le graphe doit être connexe.
 Les poids des arcs/arêtes doivent être tous différents.

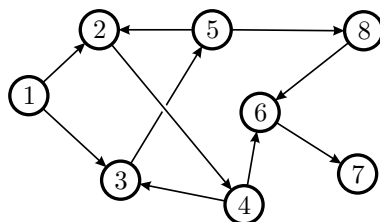


FIGURE 2 – Un graphe orienté.

27] Quelle est la longueur du plus court cycle du graphe de la FIGURE 2 ?

- 4
 5
 6
 il n'y en a pas

28] Laquelle de ces matrices est la matrice d'adjacence de la fermeture transitive réflexive du graphe de la FIGURE 2 ?

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

29] Si l'on effectue un parcours en largeur du graphe de la FIGURE 2 en partant du sommet 3, quel sommet sera le père du sommet 6 ?

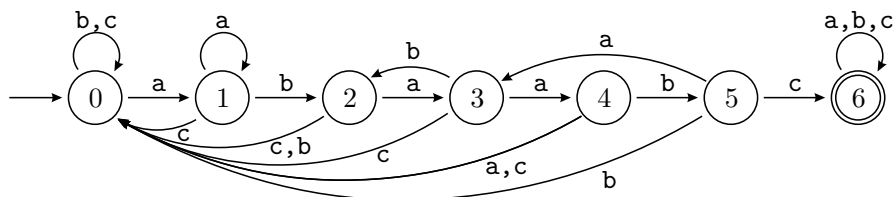
- 4
 7
 8
 cela dépend

30] Parmi les tableaux de pères suivants, lequel ne correspond pas à un recouvrement du graphe de la FIGURE 2 par un arbre ?

- [- 1 1 2 3 8 6 5]
 [- 1 4 2 3 8 6 5]
 [- 1 1 3 3 4 6 5]
 [- 5 1 2 3 4 6 5]

31] Dans une représentation par listes de successeurs d'un graphe orienté à n sommets, combien de fois au maximum un même sommet peut-il apparaître dans l'ensemble des listes de successeurs ?

- 1
 $\lceil \log n \rceil$
 $\lfloor \frac{n}{2} \rfloor$
 n



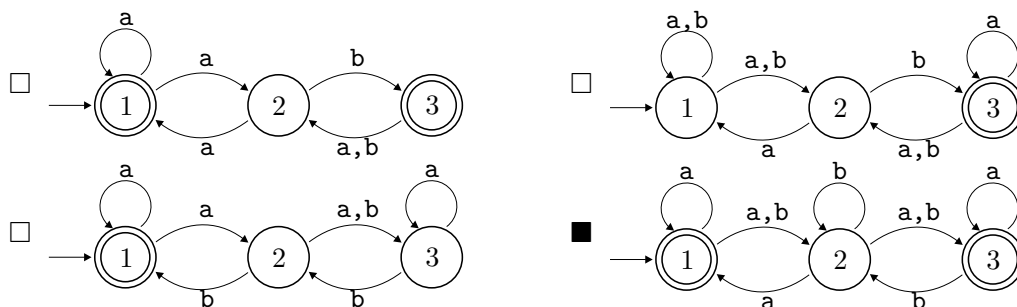
32] L'automate déterministe ci-dessus permet la recherche du motif **abaabc** dans un texte, mais il contient une erreur ! De quel état la transition erronée part-elle ?

- 2
 3
 4
 5

33] Quel est le nombre minimal d'états que peut comporter un automate *non-déterministe* pour la recherche d'un motif de longueur m dans un texte ?

- $\log m$
 $\frac{m}{2}$
 m
 $m + 1$

34] Lequel des automates non-déterministes suivants ne reconnaît pas le motif **aabab** ?



35] À quoi sert le modulo dans l'algorithme de recherche de motif de Rabin-Karp avec modulo ?

- À diminuer le coût des comparaisons.
 À diminuer le nombre de comparaisons.
 À améliorer la complexité spatiale de l'algorithme.
 À rendre l'algorithme moins sujet aux biais statistiques.

36] On implémente un dictionnaire contenant n mots/définitions à l'aide d'une table pour recherche dichotomique. Quels sont les complexité respectives des opérations de recherche d'un mot et d'insertion d'un nouveau mot dans ce dictionnaire ?

- $\Theta(1)$ et $\Theta(n)$
 $\Theta(\log n)$ et $\Theta(1)$
 $\Theta(\log n)$ et $\Theta(n)$
 $\Theta(\log n)$ chacune

37] Un algorithme de complexité exponentielle mets environ 4 secondes à traiter une entrée de taille $n = 5$. Quel temps mettra-t-il (à peu près) à traiter une entrée de taille $n = 10$?

- 8s
 2 minutes
 16s
 on manque d'information pour savoir

38] Quelle est la complexité moyenne de la suppression du i -ème élément d'une liste chaînée contenant n éléments ?

- $\Theta(1)$
 $\Theta(\log n)$
 $\Theta(n)$
 $\Theta(n \log n)$

39] En calculant la complexité d'un algorithme de type « diviser pour régner » on aboutit à la formule de récurrence : $T(n) = 3 \times T(\frac{n}{2}) + 4 \times n$. En combien de sous-problèmes le problème est-il divisé à chaque fois ?

- 2 3 4 on ne sait pas

40] Laquelle de ces affirmations concernant les arbres est fausse ?

- Une feuille possède toujours un père.
 Un nœud interne possède toujours au moins un fils.
 La racine peut-être une feuille.
 Un frère (fils du père) d'un nœud interne peut être une feuille.

41] On part d'une pile vide et on exécute la série d'instructions : `push(3)`, `push(5)`, `pop()`, `push(8)`, `pop()`, `pop()`, `push(7)`, `push(4)` et `pop()`. Quel élément reste-t-il à la fin dans la pile ?

- 3 4 5 7

42] Quelle est la complexité dans le pire cas de l'affichage par un parcours *postfixe* de l'ensemble des nœuds d'un arbre binaire contenant n éléments ?

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