

## QCM de révisions

- Le QCM ci-après compte 88 questions et est à traiter en 2h.
- Toutes les questions ont au moins un item à cocher. Certaines en ont plusieurs. Une question est réussie si tous les items corrects sont cochés et aucun item incorrect n'est coché.

### Structures de données

1. La suite de bits 11101100 peut a priori représenter ...
  - (A) Un entier naturel plus grand que 1000.
  - (B) Un entier strictement négatif.
  - (C) Un rationnel.
  - (D) Un caractère.
  - (E) Rien de ce qui précède.
2. Une pile est ...
  - (A) L'inverse d'une file.
  - (B) Une structure FIFO.
  - (C) Implémentable à l'aide d'un tableau avec des opérations en temps constant.
  - (D) Utile pour implémenter une liste doublement chaînée.
  - (E) Une structure disponible nativement en C.
  - (F) Aucune des réponses précédentes.
3. Une file peut ...
  - (A) Être implémentée par deux piles avec des opérations élémentaires en temps amorti constant.
  - (B) Être implémentée par deux files avec des opérations élémentaires en temps amorti constant.
  - (C) Être implémentée par une seule pile.
  - (D) Être implémentée par un tas.
  - (E) Simuler une table de hachage.
4. Qu'est-ce qu'une collision ?
  - (A) L'absence de réponse dans une requête serveur.
  - (B) L'arrêt d'un fil d'exécution à cause d'une impossibilité d'entrer en section critique.
  - (C) La tentative de retrait d'une pile vide.
  - (D) La tentative d'ajout dans une file pleine.
  - (E) Deux éléments ayant la même valeur de hachage.
5. Les tableaux circulaires ...
  - (A) Nécessitent une architecture spéciale.
  - (B) Permettent d'implémenter les piles.
  - (C) Permettent d'implémenter les files.
  - (D) Ont une taille fixe.
  - (E) Sont spécifiques au C et au Python et ne peuvent pas être utilisés en Ocaml.

6. Une fonction de hachage peut être utilisée ...

- (A) En sécurité informatique.
- (B) Pour la reconnaissance de motif dans un texte.
- (C) Pour la mémorisation.
- (D) Pour accélérer l'algorithme des  $k$  plus proches voisins.

7. Si on considère un arbre binaire strict et non vide à  $n$  noeuds internes,  $f$  feuilles et de hauteur  $h$ , quelles relations sont vérifiées parmi les suivantes ?

- (A)  $f \leq n$ .
- (B)  $f + n < 2^{h+1} - 1$ .
- (C)  $n \geq h$ .
- (D)  $f = n + 1$ .
- (E)  $f + n + h \neq 0$ .

8. En ignorant la valeur des étiquettes des noeuds, combien y a-t-il d'arbres binaires de hauteur exactement 2 ?

- (A) 3.
- (B) 14.
- (C) 21.
- (D) 35.
- (E) Cela dépend.

9. Le parcours en largeur d'un arbre est ...

- (A) Linéaire en le nombre de noeuds.
- (B) Linéaire en le nombre d'arêtes.
- (C) Linéaire en la hauteur.
- (D) Jamais utilisé en pratique.
- (E) L'ordre inverse du parcours en profondeur.

10. Dans un arbre binaire de recherche (ABR) de taille  $n$  et de hauteur  $h$ , on peut ...

- (A) Trouver un élément en  $O(n)$ .
- (B) Trouver un élément en  $O(h)$ .
- (C) Trouver un élément en  $O(\log n)$ .
- (D) Insérer un élément en  $O(\log n)$ .
- (E) Supprimer un élément en  $O(\log n)$ .
- (F) Vérifier qu'il est bien ABR en  $O(n)$ .

11. Les arbres rouge-noir ...

- (A) Sont des ABR.
- (B) Peuvent être utilisés pour implémenter un dictionnaire.
- (C) N'assurent aucune garantie de complexité.
- (D) N'ont pas d'utilité pratique.
- (E) Nécessitent un bit d'information supplémentaire par noeud par rapport à un arbre binaire.
- (F) Ne peuvent pas stocker de valeurs.

12. La structure de tas peut être utilisée dans les algorithmes suivants ...
- (A) Algorithme de Kruskal.
  - (B) Algorithme de parcours en profondeur.
  - (C) Algorithme de Kosaraju.
  - (D) Algorithme du tri par tas.
  - (E) Algorithme de Dijkstra.
13. Il existe une implémentation de la structure unir et trouver ayant une complexité ...
- (A) Constante pour l'opération unir.
  - (B) Constante pour l'opération trouver.
  - (C) Constante pour l'opération trouver et l'opération unir.
  - (D) Constante en complexité amortie pour l'opération trouver et l'opération unir.
  - (E) En  $O(\log n)$  dans le pire cas pour les opérations trouver et unir où  $n$  est le nombre d'éléments manipulés.
14. Un graphe est régulier si tous ses sommets ont le même degré. Existe-t-il un graphe régulier ayant ...
- (A) 9 sommets.
  - (B) Degré 5 avec 3 composantes connexes.
  - (C) Degré 5 avec 9 sommets.
  - (D) Degré 6 avec 6 sommets.
  - (E) Un degré  $p$  et un nombre  $q$  de sommets avec des  $p$  et  $q$  premiers.
15. Le nombre chromatique d'un graphe  $G$  ...
- (A) Est le plus grand entier  $k$  tel qu'il existe une  $k$ -coloration de  $G$ .
  - (B) Est le plus petit entier  $k$  tel qu'il existe une  $k$ -coloration de  $G$ .
  - (C) Dépend de l'implémentation des graphes en machine.
  - (D) Est majoré par le degré maximum du graphe.
  - (E) Est majoré par le degré minimum du graphe.
16. L'ensemble des graphes orientés à  $n$  sommets ...
- (A) Est en bijection avec l'ensemble des graphes non orientés à  $n$  sommets.
  - (B) Est en bijection avec l'ensemble des graphes non orientés à  $\lceil n/2 \rceil$  sommets.
  - (C) Est en bijection avec le groupe symétrique  $S_n$ .
  - (D) Est en bijection avec l'ensemble des mots de  $\{a, b\}^{n^2-n}$ .
  - (E) Est en bijection avec l'ensemble des langages qui contiennent  $n$  mots sur  $\{a, b\}$ .
17. Un graphe non orienté à  $m$  arêtes et  $n$  sommets est un arbre si et seulement si ...
- (A) Il est acyclique.
  - (B) Il est acyclique avec  $m = n - 1$ .
  - (C) Il possède une racine.
  - (D) Toute paire de sommets est reliée par un unique chemin.
  - (E) Il est connexe avec  $m < n$ .

18. Le poids d'un sous-graphe  $T$  de  $G$  ...
- (A) Ne peut pas être négatif.
  - (B) Est majoré par la somme des poids de toutes les arêtes de  $G$ .
  - (C) Est la somme des poids des arêtes de  $T$ .
  - (D) Dépend du sens du parcours du graphe.
  - (E) Aucune des réponses ci-dessus.
19. La complexité d'un parcours en profondeur du graphe  $G = (S, A)$  est ...
- (A) En  $O(|S|^2)$  quelle que soit la représentation du graphe.
  - (B) En  $O(|A|)$  si  $G$  représenté par listes d'adjacence.
  - (C) En  $O(|S|^2)$  si  $G$  est représenté par matrice d'adjacence.
  - (D) En  $O(|S| + |A|)$  si  $G$  est représenté par matrice d'adjacence.
  - (E) En  $O(|S| + |A|)$  si  $G$  est représenté par liste d'arêtes.
20. La représentation la plus pertinente du graphe des pages web reliées par un lien hypertexte est ...
- (A) Par liste d'arêtes.
  - (B) Par matrice d'adjacence.
  - (C) Par listes d'adjacence.
  - (D) Par forêt.
  - (E) Par tas.
21. Soit  $A$  un ensemble construit par induction. Alors ...
- (A) Tout ensemble  $E$  possédant les mêmes propriétés que  $A$  englobe  $A$ .
  - (B) Certains objets de  $A$  ont été construits par une infinité d'opérations.
  - (C)  $A$  est généralement en bijection avec  $\mathbb{N}$ .
  - (D) On peut faire une preuve par induction pour montrer que tous les objets de  $A$  vérifient une propriété.

## Algorithmique

22. Un invariant de boucle ...
- (A) Aide à prouver la correction d'un algorithme.
  - (B) Est vérifié si et seulement si la propriété reste vraie après une itération si on la suppose vraie avant.
  - (C) Est vérifié en sortie de boucle.
  - (D) Correspond à une formulation de la correction de l'algorithme sur des sous-problèmes.
  - (E) Aucune des propositions ci-dessus.
23. Un algorithme qui termine en temps probabiliste avec une réponse exacte est un algorithme de ...
- (A) Atlantic City.
  - (B) Atlanta.
  - (C) Las Vegas.
  - (D) Macao.
  - (E) Monte Carlo.

24. Un algorithme dont le temps de calcul est garanti mais dont le résultat peut être inexact avec une certaine probabilité est un algorithme de ...
- (A) Atlantic City.
  - (B) Atlanta.
  - (C) Las Vegas.
  - (D) Macao.
  - (E) Monte Carlo.
25. Si pour un problème  $\mathcal{P}$  on dispose d'un algorithme  $\mathcal{A}$  qui s'exécute en temps polynomial, sans faux négatif et avec probabilité de faux positif inférieure à  $1/3$  ...
- (A)  $\mathcal{P} \in P$ .
  - (B) On peut résoudre  $\mathcal{P}$  sans faux négatif et avec une probabilité de faux positif arbitrairement petite.
  - (C) Si  $\mathcal{A}$  affirme qu'une instance de  $\mathcal{P}$  est négative, c'est bien le cas.
  - (D) Il existe un algorithme qui résout  $\mathcal{P}$  en temps polynomial en moyenne.
26. L'algorithme du tri rapide ...
- (A) A une complexité pire cas en  $\Theta(n \log n)$  et une complexité moyenne en  $\Theta(n)$ .
  - (B) A une complexité pire cas en  $\Theta(n^2)$  et une complexité moyenne en  $\Theta(n \log n)$ .
  - (C) Est un algorithme de type Las Vegas.
  - (D) A la même complexité dans le pire cas et en moyenne.
  - (E) Peut terminer plus tard dans sa version probabiliste que dans le pire cas de la méthode déterministe.
27. Un algorithme par retour sur trace (backtracking) ...
- (A) Correspond à un parcours en largeur.
  - (B) Correspond à un parcours en profondeur.
  - (C) Peut entraîner des boucles.
  - (D) Nécessite d'organiser les données.
  - (E) Permet une résolution polynomiale d'un Sudoku.
28. Un algorithme de type branch-and-bound ...
- (A) S'applique à un problème de décision.
  - (B) Est une variante d'algorithme diviser pour régner.
  - (C) Est une variante d'un algorithme de backtracking.
  - (D) Renvoie une approximation de la solution.
  - (E) Est utile pour trouver un chemin dans un graphe.
  - (F) Aucune des réponses précédentes.
29. Un algorithme glouton ...
- (A) N'est utile que s'il donne une solution exacte.
  - (B) Peut être un algorithme d'approximation.
  - (C) Est forcément polynomial.
  - (D) Est un cas particulier de backtracking.
  - (E) Ne revient jamais sur une décision prise à une étape précédente.

30. Quels sont les algorithmes gloutons parmi les suivants ?

- (A) Algorithme de Huffman.
- (B) Algorithme de Dijkstra.
- (C) Algorithme de Kruskal.
- (D) Algorithme pour sac à dos en prenant les objets par ordre croissant de valeur.

31. On note  $(I_n)_{n \in \mathbb{N}}$  une suite d'instances d'un problème de minimisation  $\mathcal{P}$ ,  $\mathcal{A}$  un algorithme qui donne des solutions aux instances de  $\mathcal{P}$ ,  $C_n^*$  le coût optimal pour  $I_n$  et  $C_n$  le coût de la solution trouvée par  $\mathcal{A}$  sur  $I_n$ .

- (A) Pour tout  $n \in \mathbb{N}$ ,  $C_n \leq C_n^*$ .
- (B) S'il existe  $\alpha > 0$  tel que  $\forall n \in \mathbb{N}$ ,  $C_n \leq \alpha C_n^*$ , alors  $\mathcal{A}$  est une  $\alpha$ -approximation pour  $\mathcal{P}$ .
- (C) Si  $C_n^*/C_n \rightarrow +\infty$  à l'infini, alors  $\mathcal{A}$  n'est pas un algorithme d'approximation pour  $\mathcal{P}$ .
- (D) Si  $C_n/C_n^* \rightarrow +\infty$  à l'infini, alors  $\mathcal{A}$  n'est pas un algorithme d'approximation pour  $\mathcal{P}$ .

32. Un algorithme de programmation dynamique ...

- (A) Nécessite de stocker des valeurs dans un tableau.
- (B) Consiste exclusivement à mémoriser des résultats.
- (C) Consiste à partitionner les solutions en sous-problèmes distincts.
- (D) Consiste à formuler la solution d'un problème en fonction de la solution à des sous problèmes.
- (E) Est inadapté lorsque les sous problèmes se recourent.

33. Dans un algorithme de type diviser pour régner ...

- (A) La phase de division coupe le problème en deux sous-problèmes.
- (B) La phase de résolution des sous-problèmes se fait généralement de manière itérative.
- (C) La phase de combinaison a toujours une complexité négligeable par rapport aux deux phases précédentes.
- (D) Ne donne que des algorithmes de complexité supérieure à un  $\Theta(n)$ .
- (E) Aucune des réponses précédentes.

34. En posant  $C(0) = C(1) = 1$ , pour quelles formules de récurrence obtient-on  $C(n) = \Theta(\lambda^n)$  avec  $\lambda > 1$  ?

- (A)  $C(n) = 3C(n-1)$ .
- (B)  $C(n) = \sum_{i=0}^{n-1} C(i)$ .
- (C)  $C(n) = 3C(n/2) + O(n^2 \log n)$ .
- (D)  $C(n) = nC(n/2)$ .
- (E)  $C(n) = 2C(n/2) + \Theta(2^n)$ .

35. Pour rechercher si un motif  $m$  apparaît dans un texte  $t$  on peut ...

- (A) Utiliser un automate des occurrences.
- (B) Utiliser l'algorithme de Lempel-Ziv-Welch.
- (C) Utiliser l'algorithme de Rabin-Karp.
- (D) Utiliser l'algorithme de Boyer-Moore.
- (E) Conclure en temps  $O(|m||t|)$ .

36. L'algorithme de Kosaraju appliqué à un graphe ...

- (A) Calcule les composantes fortement connexes.
- (B) Utilise un double parcours en profondeur.
- (C) Utilise le graphe transposé.
- (D) Permet de résoudre 2SAT en temps polynomial.
- (E) Utilise une structure de pile.

37. Lesquels de ces algorithmes permettent de calculer la plus petite distance entre deux sommets d'un graphe ?

- (A)  $A^*$ .
- (B) L'algorithme alpha-beta.
- (C) L'algorithme de Dijkstra.
- (D) L'algorithme de Boyer-Moore.
- (E) L'algorithme de Floyd-Warshall.

38. L'algorithme  $A^*$  avec une heuristique  $h$  ...

- (A) Correspond à l'algorithme de Floyd-Warshall si  $h$  est identiquement nulle.
- (B) Trouve un plus court chemin si  $h$  est admissible.
- (C) Trouve un plus court chemin si  $h$  est monotone.
- (D) Est polynomial en temps si  $h$  est admissible.
- (E) Est polynomial en temps si  $h$  est monotone.
- (F) N'est pas intéressant à appliquer sur un graphe non pondéré.

39. Quelles propriétés sont justes sur les arbres couvrants parmi les suivantes ?

- (A) Tout graphe possède un arbre couvrant.
- (B) L'arête de poids minimal appartient à un arbre couvrant de poids minimal.
- (C) L'arête de poids maximal n'appartient à aucun arbre couvrant minimal.
- (D) Si les poids de toutes les arêtes de  $G$  sont deux à deux distincts et que  $G$  est connexe alors  $G$  admet un unique arbre couvrant minimal.
- (E) Si  $G$  admet un unique arbre couvrant de poids minimal, les poids de ses arêtes sont deux à deux distincts.

40. Lesquelles parmi les propriétés suivantes sur les couplages sont vraies ?

- (A) Un couplage maximal est maximum.
- (B) Un couplage maximum est maximal.
- (C) Un graphe admettant un couplage parfait est nécessairement biparti.
- (D) Dans un graphe biparti, on peut calculer un couplage maximal en temps polynomial.
- (E) Un couplage sans chemin augmentant dans un graphe biparti est maximum.

41. L'algorithme des  $k$  plus proches voisins ...

- (A) Est un algorithme d'apprentissage non supervisé.
- (B) Gagne en précision lorsqu'on augmente  $k$ .
- (C) Ne peut pas être accéléré par pré-traitement.
- (D) Résiste au fléau de la dimension.
- (E) Aucune des réponses ci-dessus.

42. Un arbre de décision ...
- (A) Peut être construit par un algorithme d'apprentissage supervisé.
  - (B) Classe toujours correctement tous les éléments de l'ensemble d'apprentissage.
  - (C) Peut être amélioré à l'aide d'arbres k-d.
  - (D) Peut être calculé sans calcul d'entropie par l'algorithme ID3.
  - (E) A une hauteur bornée par la dimension des données.
43. L'algorithme des  $k$  moyennes ...
- (A) Repose sur l'entropie de Shannon.
  - (B) Converge vers une réponse optimale.
  - (C) Ne converge pas nécessairement.
  - (D) Ne reconnaît pas les classes non convexes.
  - (E) Nécessite un calcul de médioïde.
44. La classification obtenue par un algorithme de regroupement hiérarchique ascendant ...
- (A) Est indépendante de la distance choisie.
  - (B) Permet de choisir le nombre de clusters.
  - (C) Ne reconnaît pas les classes non convexes.
  - (D) Nécessite un calcul de centroïde.
  - (E) A une complexité dépendante de la distance considérée.
45. Un algorithme reposant sur une heuristique ...
- (A) Ne peut pas donner une réponse optimale à coup sûr.
  - (B) Ne peut généralement pas garantir le temps d'exécution et l'optimalité du résultat.
  - (C) Ne peut pas être exécuté avec une heuristique différente.
  - (D) Peut être ajusté pour une utilisation pratique précise en adaptant l'heuristique considérée.
  - (E) Aucune des réponses ci-dessus.
46. Les positions gagnantes pour un joueur  $J_1$  dans un jeu à deux joueurs ...
- (A) Peuvent être calculées en temps polynomial en la taille du jeu.
  - (B) Dépendent des réponses de l'adversaire.
  - (C) Sont nécessairement des sommets contrôlés par le joueur  $J_1$ .
  - (D) Peuvent mener à une défaite de  $J_1$ .
  - (E) Aucune des réponses ci-dessus.
47. L'algorithme de Peterson implémente un verrou qui vérifie ...
- (A) L'exclusion mutuelle.
  - (B) La bijection des traces d'exécution.
  - (C) L'absence de famine.
  - (D) La faible consommation de ressources par les fils.
  - (E) La correction du programme qui l'utilise.
  - (F) Aucune des réponses précédentes.



## Logique

48. Quelles formules sont des tautologies ?

(A)  $((A \Rightarrow B) \Rightarrow C) \Leftrightarrow (A \Rightarrow (B \Rightarrow C))$ .

(B)  $(A \vee B) \wedge (\neg A \vee \neg B)$ .

(C)  $\neg(A \wedge (A \vee B)) \Leftrightarrow \neg A$ .

(D)  $((A \vee B) \wedge (C \vee D)) \Leftrightarrow (A \wedge B) \vee (B \wedge D)$ .

(E)  $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$ .

49. La taille d'une formule est majorée par sa hauteur.

(A) Vrai.

(B) Faux.

50. Une variable libre ...

(A) Peut être liée à un autre endroit dans la formule.

(C) N'influe pas la sémantique d'une formule.

(B) A une portée.

(D) Est toujours associée à une variable liée.

51. Soit  $\varphi$  et  $\psi$  deux formules telles que  $\psi \models \varphi$ . Cela signifie ...

(A) Que  $\varphi$  est conséquence de  $\psi$ .

(D) Si  $\varphi$  est une antilogie, alors  $\psi$  aussi.

(B) Que  $\psi$  est conséquence de  $\varphi$ .

(E) Cela n'a pas de sens, la notation  $\models$  n'existe que sous la forme  $v \models \varphi$  avec  $v$  une valuation.

(C) Que les valuations satisfaisant  $\varphi$  satisfont aussi  $\psi$ .

52. Une équivalence entre formules du calcul propositionnel ...

(A) Signifie qu'elles ont le même arbre de dérivation.

(B) Peut se prouver à l'aide de tables de vérité.

(C) Peut se prouver à l'aide de substitutions dans des formules qu'on sait déjà être équivalentes.

(D) Ne peut pas être démontrée sans la déduction naturelle.

(E) Peut se prouver par une étude de la matrice de confusion.

53. Pour toute formule du calcul du calcul propositionnel il existe une formule équivalente ...

(A) Sous forme normale conjonctive.

(D) Sous forme normale littérale.

(B) Sous forme normale disjonctive.

(E) Sous forme normale de Chomsky.

(C) Sous forme normale de Quine.

(F) Aucune des réponses précédentes.

54. On sait résoudre le problème SAT en temps polynomial sur des instances ...

(A) Sous 2-CNF.

(D) Sous 3-DNF.

(B) Sous 3-CNF.

(E) Ne faisant intervenir que les connecteurs  $\wedge$  et  $\neg$ .

(C) Sous 2-DNF.

55. Un séquent est constitué de prémisses et d'une conclusion.

- (A) Vrai. (B) Faux.

56. Un système de preuve qui permet de prouver au moins un séquent ...

- (A) Contient au moins un axiome. (D) Contient les règles de la déduction naturelle.  
(B) Correspond à une règle d'inférence. (E) Est une famille de séquents.  
(C) Permet de construire des arbres de preuve.

57. Quels séquents sont prouvables en logique classique ?

- (A)  $\neg\neg A \vdash A$ . (D)  $A \vee B \vdash A \wedge B$ .  
(B)  $\neg\neg\neg A \vdash \neg A$ . (E)  $A, A \rightarrow B \vdash B$ .  
(C)  $A \wedge B \vdash A \vee B$ .

58. Pour la déduction naturelle en logique propositionnelle,  $\Gamma \vdash F$  si et seulement si  $\Gamma \models F$ .

- (A) Vrai. (B) Faux.

## Théorie des langages

59. Si  $u$  et  $v$  sont deux mots, quelles propriétés parmi les suivantes sont vraies ?

- (A)  $|u| > 0$ . (D) Si  $u = vw$  alors  $v$  est un sous-mot de  $u$ .  
(B)  $u$  est préfixe de  $v$  ou  $v$  est préfixe de  $u$ . (E) Si  $u = vw$  alors  $v$  est un facteur de  $u$ .  
(C) Si  $u$  est préfixe de  $v$  alors  $v$  est suffixe de  $u$ .

60. Si  $L$ ,  $L'$  et  $L''$  sont des langages alors ...

- (A) Si  $L$  et  $L'$  sont finis,  $|LL'| = |L| \times |L'|$ . (D)  $L(L' \cap L'') = LL' \cap LL''$ .  
(B)  $L^*$  est infini. (E) Aucune des réponses précédentes.  
(C)  $L \subset LL$ .

61. Si  $L$  et  $L'$  sont deux langages tels que  $L \subset L'$ , quelles affirmations sont vraies ?

- (A) Si  $L'$  est reconnaissable alors  $L$  aussi.  
(B) Si  $L'$  n'est pas rationnel,  $L$  non plus.  
(C) Si  $L'$  est fini,  $L$  est rationnel.  
(D) Si  $L$  est reconnaissable, il l'est par un automate déterministe et complet.  
(E) Si  $L$  est reconnaissable, il l'est par un automate émondé et complet.

62. Le langage  $L = \{a^n b^m \mid n \equiv m \pmod{2}\}$  est dénoté par l'expression régulière ...
- (A)  $(aa + bb)^*$ . (D)  $a^n b^m$ .  
 (B)  $(aa + ab + ba + bb)^*$ . (E) Aucune des réponses précédentes.  
 (C)  $((a(a + b)^* b)^* + (b(a + b)^* a)^*)^*$ .
63. Les langages rationnels sont stables par ...
- (A) Préfixe. (C) Complémentaire.  
 (B) Miroir. (D) Intersection.
64. Un automate fini déterministe et complet ...
- (A) Est nécessairement émondé.  
 (B) Est unique pour chaque langage à renommage près des états.  
 (C) Permet de tester l'appartenance d'un mot à un langage en temps linéaire.  
 (D) Donne aisément un automate reconnaissant le complémentaire du langage.  
 (E) Est de taille exponentielle en la taille d'un automate non déterministe reconnaissant le même langage.
65. Si  $e$  est une expression rationnelle, le problème consistant à savoir si un mot  $u$  appartient à  $L(e)$  ...
- (A) Nécessite au moins une complexité linéaire en  $|u|$ . (D) N'a pas d'application pratique.  
 (B) Peut être résolu en temps polynomial en  $|u| + |e|$ . (E) Aucune des réponses précédentes.  
 (C) Est un problème indécidable.
66. On peut montrer que tout langage rationnel est reconnaissable par un automate fini ...
- (A) En construisant l'automate des parties. (D) Avec l'algorithme de McNaughton Yamada.  
 (B) En utilisant les automates de Thompson. (E) En utilisant l'algorithme de Berry-Sethi.  
 (C) En utilisant la méthode d'élimination des états.
67. Parmi les langages suivants, lesquels sont rationnels ?
- (A)  $\{a^n b^n \mid n \geq 0\}$ . (D) Les mots sur  $\{0, 1\}$  correspondant aux écritures binaires d'entiers divisibles par 3.  
 (B) Les mots bien parenthésés.  
 (C)  $\{a^n b^m \mid n \equiv m \pmod{2}\}$ . (E)  $\{a^p \mid p \text{ est premier}\}$ .
68. Parmi les conditions suivantes, lesquelles sont suffisantes pour que  $L$  soit rationnel ?
- (A)  $L$  est dénombrable. (D)  $L^*$  est rationnel.  
 (B)  $L$  est reconnu par un automate non déterministe. (E)  $L \subset \{a\}^*$ .  
 (C)  $L$  est le complémentaire d'un langage rationnel. (F)  $L$  est local.

69. Si un langage vérifie le lemme de l'étoile, alors il est rationnel.

(A) Vrai.

(B) Faux.

70. Une dérivation  $u \Rightarrow v$  est immédiate si ...

(A) Tous les non terminaux de  $u$  sont dérivés.

(D)  $v$  ne contient plus de non terminal.

(B) Un seul non terminal de  $v$  est dérivé.

(E)  $u$  est l'axiome.

(C) Un seul non terminal de  $u$  est dérivé.

71. Les langages algébriques ...

(A) Sont inclus dans les langages rationnels.

(D) Sont stables par intersection.

(B) Sont stables par étoile de Kleene.

(E) Sont stables par concaténation.

(C) Sont stables par complémentaire.

72. Quelles sont les grammaires ambiguës ?

(A)  $S \rightarrow SS \mid a$ .

(D)  $S \rightarrow aSb \mid ab \mid \varepsilon$ .

(B)  $S \rightarrow ST \mid TS, T \rightarrow SS$ .

(E)  $S \rightarrow aS \mid bS \mid \varepsilon$ .

(C)  $S \rightarrow aSbS \mid bSaS \mid \varepsilon$ .

73. Les dérivations droites sont en bijection avec les arbres de dérivation.

(A) Vrai.

(B) Faux.

74. Un analyseur lexical ...

(A) Intervient avant l'analyseur syntaxique.

(C) Détermine la structure d'un programme.

(B) Identifie les lexèmes.

(D) Peut utiliser un automate fini.

75. Un analyseur syntaxique ...

(A) Ne peut gérer que des grammaires non ambiguës.

(B) Considère les lexèmes comme des terminaux.

(C) N'est pas utilisé en pratique.

(D) Choisit la grammaire à considérer en fonction des données en entrée.

(E) Aucune des réponses précédentes.

76. Savoir si un mot est engendré par une grammaire non contextuelle est un problème indécidable.

(A) Vrai.

(B) Faux.

77. Savoir si une grammaire non contextuelle est ambiguë est un problème indécidable.

(A) Vrai.

(B) Faux.

## Complexité et calculabilité

78. Parmi les problèmes ci-dessous, lesquels sont des problèmes de décision ?

(A) La résolution d'une grille de Sudoku.

(D) Le problème de l'arrêt.

(B) La primalité d'un entier.

(E) L'égalité entre deux langages.

(C) Le plus court chemin entre deux points.

79. Parmi les problèmes ci-dessous, lesquels sont des problèmes d'optimisation ?

(A) La résolution d'une grille de Sudoku.

(D) Le problème de l'arrêt.

(B) La primalité d'un entier.

(E) L'égalité entre deux langages.

(C) Le plus court chemin entre deux points.

80. Si  $n$  est la taille de l'entrée, pour quelles complexités peut-on dire que l'algorithme considéré est polynomial ?

(A)  $O(\log n!)$ .

(D)  $O(\sqrt{n})$ .

(B)  $2^{o(n)}$ .

(E)  $O(2^n)$ .

(C)  $O((\log n)^n)$ .

81. L'algorithme cherchant à déterminer si un entier  $n$  est premier en calculant son modulo par tous les entiers entre 2 et  $\lceil \sqrt{n} \rceil$  est de complexité ...

(A) Quasi-linéaire.

(D) Exponentielle.

(B) Polynomiale.

(E) Linéaire.

(C) Sous-linéaire.

82. La complexité d'un problème de décision correspond à ...

(A) La complexité pire cas du meilleur algorithme qui le résout.

(B) La complexité pire cas du pire algorithme qui le résout.

(C) La complexité meilleur cas du meilleur algorithme qui le résout.

(D) Dépend du format d'entrée des instances.

(E) N'est pas définie.

83. Si le problème  $A$  se réduit polynomialement au problème  $B$  alors ...
- (A) Si  $A$  se résout en temps polynomial,  $B$  aussi.
  - (B) Si  $B$  se résout en temps polynomial,  $A$  aussi.
  - (C)  $A$  est considéré comme plus difficile que  $B$ .
  - (D) Une instance de  $A$  peut être transformée en une instance de  $B$ .
  - (E) Si  $A$  est décidable alors  $B$  aussi.
84. Si un problème est dans NP alors ...
- (A) On ne sait pas le résoudre en temps polynomial.
  - (B) Il peut être résolu en temps exponentiel.
  - (C) Il peut être résolu en espace polynomial.
  - (D) Il s'agit d'un problème fonctionnel.
  - (E) Pour toute instance du problème, on peut créer un certificat en temps polynomial.
85. En supposant que  $P \neq NP$ , le problème du voyageur de commerce ...
- (A) Est NP-complet dans sa version classique et dans P dans sa version euclidienne.
  - (B) Est NP-complet dans les deux versions précédemment citées.
  - (C) Admet une 2-approximation.
  - (D) N'admet aucune approximation à facteur constant.
  - (E) Peut être résolu en espace polynomial.
86. Lesquels de problèmes suivants sont NP-complets ?
- (A) 3SAT.
  - (B) Le problème de l'arrêt.
  - (C) L'existence d'un chemin hamiltonien.
  - (D) L'appartenance d'un mot à un langage régulier.
  - (E) Le problème du sac à dos.
87. Les problèmes indécidables ...
- (A) Ne sont pas les mêmes suivant le langage de programmation choisi.
  - (B) Ne peuvent pas être réduits à un autre problème.
  - (C) Dépendent de la façon dont sont représentées les instances du problème.
  - (D) Ne concernent que des problèmes se rapportant au fonctionnement d'algorithmes.
  - (E) Aucune des réponses précédentes.
88. Parmi les suivants, quels sont les problèmes indécidables ?
- (A) Le problème de l'arrêt sur les entrées de taille inférieure à 5.
  - (B) Le problème de la terminaison en moins de  $n$  étapes sur toute instance de taille inférieure à  $n$ .
  - (C) Le problème de la terminaison en strictement plus de  $n$  étapes sur toute instance de taille inférieure à  $n$ .
  - (D) Le problème de l'équivalence de deux expressions régulières.
  - (E) Le problème de l'équivalence entre deux formules du calcul propositionnel.