

Il peut y avoir plusieurs items à cocher par question. Toutes les questions ont au moins un item à cocher. Une réponse est considérée comme juste si aucune mauvaise proposition est sélectionnée et que toutes les bonnes propositions sont sélectionnées.

1 Structure de Données

1. Quelle structure de donnée correspond au principe LIFO ?

- ☒ (A) Une pile
- (B) Un tableau dynamique
- (C) Une file
- (D) Une table de hachage
- (E) Une liste doublement chaînée

2. Quelle structure de donnée correspond au principe FIFO ?

- (A) Une pile
- (B) Un tableau dynamique
- ☒ (C) Une file
- (D) Une table de hachage
- (E) Une liste doublement chaînée

3. Qu'est-ce qu'une collision ?

- (A) L'absence de réponses lors d'une requête serveur.
- (B) L'arrêt d'un fil d'exécution à cause d'une impossibilité d'entrer en section critique.
- ☒ (C) Deux éléments ayant la même valeur de hachage.
- (D) La tentative de retrait d'un élément dans une pile vide.
- (E) La tentative d'ajout d'un élément dans une file pleine.

4. Une file peut :

- ☒ (A) Être implémentée par deux piles avec des opérations élémentaires en temps constant en complexité amortie.
- (B) Être implémentée par deux piles avec des opérations élémentaires en temps constant systématiquement.
- (C) Être implémentée avec une seule pile.
- (D) Simuler une table de hachage.
- (E) On peut modéliser une pile avec deux files avec des opérations élémentaires en temps constant en complexité amortie.

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 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. On considère un arbre binaire strict, avec n nœuds internes, f feuilles et hauteur h . Quelles relations sont vérifiées ?

- (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. Combien d'arbres binaires ont 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 nœuds.
- ☒ (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. On peut reconstruire un arbre binaire strict à partir du résultat de :

- (A) son parcours en largeur
- (B) son parcours en profondeur infixe

- ☒ (C) son parcours en profondeur préfixe
- ☒ (D) son parcours en profondeur postfixe

11. Un arbre bicolore est un arbre binaire de recherche.

- ☒ (A) Vrai

- (B) Faux

12. Les arbres bicolores

- ☒ (A) Peuvent être utilisés pour coder une structure de dictionnaire.
- (B) N'assurent aucune garantie de complexité
- (C) N'ont pas d'utilité pratique.

- ☒ (D) Nécessitent un bit d'informations supplémentaires par nœuds qu'un arbre binaire.
- (E) Ne peuvent pas stocker de valeurs.

13. La structure de tas est utilisée dans les algorithmes suivants :

- ☒ (A) Algorithme de Huffman
- (B) Algorithme de parcours en profondeur
- (C) Algorithme de Kosaraju

- ☒ (D) Algorithme de tri par tas
- ☒ (E) Algorithme de Dijkstra

14. Un tas à n noeuds est

- ☒ (A) Un arbre binaire parfait.
- ☒ (B) Avec $\lceil \frac{n}{2} \rceil$ feuilles.
- ☒ (C) De hauteur $O(\log_2(n))$.

- ☒ (D) Représentable facilement par un tableau de taille n .
- (E) Un arbre binaire de recherche équilibré.

15. Il existe une implémentation de la structure unir et trouver qui permet une complexité :

- ☒ (A) Constante pour l'opération unir
- ☒ (B) Constante pour l'opération trouver
- (C) Constante dans le pire cas pour l'opération unir et l'opération trouver.

- (D) Constante en complexité amortie pour l'opération unir et l'opération trouver.
- (E) En $o(\log(n))$ dans le pire cas pour les opérations unir et trouver.

16. Un graphe est régulier si tous ses sommets ont même degré. Existent-ils des graphes réguliers ayant les propriétés suivantes ?
- (A) Avoir 9 sommets
 (B) De degré 5 avec 3 composantes connexes.
 (C) De degré 5 avec 9 sommets
 (D) De degré 6 avec 6 sommets
 (E) Avec un degré p et un nombre de sommets q tels que p et q sont premiers.
17. Le nombre chromatique d'un graphe G
- (A) Correspond au plus grand entier k tel qu'il existe une k -coloration de G .
 (B) Correspond au plus petit entier k tel qu'il existe une k -coloration de G .
 (C) Dépend de l'implémentation des graphes en machines.
 (D) Est majoré par le degré maximum du graphe.
 (E) Est minoré par le degré minimum du graphe.
18. 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 $\lceil \frac{n}{2} \rceil$ sommets
 (C) Est en bijection avec l'ensemble des mots de $\{a; b\}^{n^2-n}$
 (D) Est en bijection avec l'ensemble des langages sur l'alphabet $\{a; b\}$ à n lettres.
 (E) Est en bijection avec le groupe symétrique S_n .
19. Un graphe non orienté à m arêtes sur n sommets est un arbre si et seulement si :
- (A) Il est acyclique.
 (B) Il est acyclique avec $m = n - 1$.
 (C) Toute paire de sommets est reliée par un unique chemin.
 (D) Il possède une racine.
 (E) Il est connexe avec $m < n$.
20. Le poids d'un sous-graphe T de G
- (A) Ne peut pas être négatif.
 (B) Est majoré par la somme du poids de toutes les arêtes du graphe G .
 (C) Correspond à la somme des poids des arêtes de T .
 (D) Dépend du sens de parcours du graphe.
 (E) Aucune des réponses ci-dessus.
21. La complexité d'un parcours en profondeur d'un graphe $G = (S, A)$ est :
- (A) En $O(|S|^2)$ quelle que soit la représentation du graphe.
 (B) En $O(|A|)$ pour un graphe sous forme de liste d'adjacence.
 (C) En $O(|S|^2)$ pour une représentation sous forme de matrice d'adjacence.
 (D) En $O(|S| + |A|)$ pour une représentation par matrice d'adjacence.
 (E) Est en $O(|S| + |A|)$ pour une représentation par matrice d'incidence.
22. La représentation naturelle du graphe des pages webs reliés par un lien hypertexte est :
- (A) Par matrice d'incidence.
 (B) Par matrice d'adjacence.
 (C) Par liste d'adjacence.
 (D) Par forêt.
 (E) Aucune des propositions ci-dessus.
23. Soit A un ensemble construit par induction.

- (A) Tout ensemble E possédant les mêmes propriétés inductives que A englobe E .
 (B) Certains objets de A ont été construits par une infinité d'opérations.
 (C) A est en bijection avec $\mathbb{N}$.

2 Algorithmie

24. Un invariant de boucle

- (A) Permet de prouver la correction d'un algorithme.
 (B) Est vérifié si et seulement si la propriété est vraie lorsqu'on la suppose vraie au début de la boucle.
 (C) Peut ne pas être vérifié à la sortie de la boucle.
 (D) Correspond à une formulation de la correction de l'algorithme sur des sous-problèmes.
 (E) Aucune des propositions ci-dessus.

25. Un algorithme qui se termine en un temps probabiliste, mais avec une réponse exacte est un algorithme de

- (A) Atlanta
 (B) Bellagio
 (C) Las Vegas
 (D) Macao
 (E) Monte-Carlo

26. Un algorithme dont le temps de calcul est garanti, mais dont le résultat n'est correct qu'avec une certaine probabilité est un algorithme de

- (A) Atlanta
 (B) Bellagio
 (C) Las Vegas
 (D) Macao
 (E) Monte-Carlo

27. L'algorithme du tri rapide

- (A) A une complexité dans le pire cas en $\Theta(n \log(n))$ et une complexité moyenne en $\Theta(n)$.
 (B) A une complexité dans le pire cas en $\Theta(n^2)$ et une complexité moyenne en $\Theta(n \log(n))$.
 (C) Est un algorithme d'Atlanta.
 (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.

28. Parmi les algorithmes suivants lesquels permettent de trouver les occurrences d'une chaîne de caractères dans une autre :

- (A) Huffman
 (B) Bellman Ford
 (C) Boyer Moore
 (D) Rabin Karp
 (E) Liv Zempel Welch

29. Un algorithme par retour sur trace :

- (A) Correspond à un parcours en largeur.
 (B) Correspond à un parcours en profondeur.
 (C) Peut entraîner des boucles.
 (D) Nécessite d'ordonner les données.
 (E) Permet une résolution polynomiale d'un Sudoku.

30. Un parcours en profondeur permet de :

Non
pas poly

- ☒ (A) Décider si un graphe est 2-coloriable.
- ☐ (B) Calculer les distances entre chaque sommet et un point donné.
- ☒ (C) Calculer les composantes connexes.
- ☐ (D) Obtenir un tri topologique.
- ☐ (E) Détecter les cycles.
- ☒ (F) Calculer les composantes fortement connexes.

31. Un algorithme par séparation et évaluation

- ☒ (A) Permet de résoudre certains problèmes entiers en se basant sur une résolution fractionnaire.
- ☐ (B) Utilise une heuristique.
- ☒ (C) Dans un problème de minimisation consiste à minorer la valeur des solutions partielles.
- ☐ (D) Dans un problème de minimisation consiste à majorer la valeur des solutions partielles.
- ☐ (E) Nous donne un algorithme polynomial pour résoudre MAX-2-SAT

32. Quels algorithmes sont des algorithmes gloutons ?

- ☒ (A) Algorithme de Huffman
- ☒ (B) Algorithme de Dijkstra
- ☒ (C) Algorithme de Kruskal
- ☐ (D) Algorithme de résolution du sac à dos fractionnaire.
- ☒ (E) Algorithme d'ordonnancement des tâches de durée unitaire.

33. Un algorithme glouton nous donne une 2approximation au problème de couverture de sommets.

- ☒ (A) Vrai
- ☐ (B) Faux

34. Un algorithme de programmation dynamique

- ☐ (A) Nécessite de stocker des valeurs dans un tableau.
- ☒ (B) A une complexité souvent exponentielle lorsque appliqué à des problèmes sur des entrées entières.
- ☐ (C) Consiste à partitionner les solutions en sous-problèmes disjoints.
- ☒ (D) Consiste à formuler la solution d'un problème en fonction de la solution à des sous-problèmes.
- ☐ (E) Ne permet pas de résoudre un problème d'optimisation associé à un problème de décision résolu par programmation dynamique.

35. Dans un algorithme de type diviser pour régner :

- ☐ (A) La phase de division coupe le problème en deux sous-problèmes.
- ☐ (B) La phase de règne se fait généralement de façon itérative.
- ☐ (C) La phase de combinaison a une complexité toujours négligeable comparée aux phases précédentes.
- ☐ (D) Ne donne que des algorithmes de complexité supérieure à $\Theta(n)$.
- ☒ (E) Aucune des réponses ci-dessus.

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

- ☒ (A) $C(n) = 3C(n-1)$
- ☐ (B) $C(n) = \sum_{i=0}^n C(i)$
- ☐ (C) $C(n) = 3C\left(\frac{n}{2}\right) + O(n^2 \log(n))$
- ☐ (D) $C(n) = 2C\left(\frac{n}{2}\right) + O(n)$
- ☒ (E) $C(n) = 2C\left(\frac{n}{2}\right) + \Theta(2^n)$

37. L'algorithme de Kosaraju appliqué à un graphe

- (A) Calcule les composantes fortement connexes. (D) Permet de résoudre 2-SAT en temps polynomial.
 (B) Utilise un double parcours en profondeur.
 (C) Utilise le graphe transposé. (E) Utilise une structure de pile.
38. Lesquels de ces algorithmes permettent de calculer la plus petite distance entre deux points ?
 (A) A^* (D) L'algorithme de Boyer-Moore
 (B) L'algorithme alpha-beta
 (C) L'algorithme de Dijkstra (E) L'algorithme de Floyd-Warshall
39. Quelles propriétés sont justes sur les arbres couvrants ?
 (A) Tout graphe possède un arbre couvrant. minimum, les poids de ses arêtes sont distincts deux à deux.
 (B) Si le poids de toutes les arêtes de G sont deux à deux distincts et que G est connexe, G admet un unique arbre couvrant de poids minimum. (D) L'arête de poids minimum appartient à un arbre couvrant de poids minimum.
 (C) Si G admet un unique arbre couvrant de poids minimum, les poids de ses arêtes sont distincts deux à deux. (E) L'arête de poids maximum n'appartient à aucun arbre couvrant de poids minimum.
40. Quelles propositions sont vraies sur les couplages ?
 (A) Un couplage maximal est nécessairement maximum. nécessairement biparti.
 (B) Un couplage maximum est nécessairement maximal. (D) Tout graphe biparti admet un nombre pair de sommets.
 (C) Un graphe admettant un couplage parfait est nécessairement biparti. (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. (C) Ne peut être accéléré par pré-traitement.
 (B) Gagne en précision lorsqu'on augmente k . (D) Nécessite que k soit impair.
 (E) Aucune des réponses ci-dessus.
42. Un arbre de décision
 (A) Est un algorithme d'apprentissage supervisé (D) Peut être construit sans calcul d'entropie grâce à l'algorithme ID3
 (B) Classe correctement tous les éléments de l'ensemble d'apprentissage (E) A une hauteur bornée par la dimension des données.
 (C) Peuvent être améliorés à l'aide d'arbres k-d.
43. Algorithme des k-moyennes
 (A) Repose sur l'entropie de Shanon (D) Ne reconnaît pas des classes non-convexes
 (B) Converge vers une réponse optimale.
 (C) Ne converge pas nécessairement. (E) Nécessite un calcul de médoid.
44. La classification obtenue par un algorithme de regroupement hiérarchique ascendant :

- (A) Est indépendante de la distance choisie.
- (B) Ne permet pas "la remise en question".
- (C) A une complexité dépendante de la complexité du calcul de la distance considérée.
- (D) Ne reconnaît pas des classes non-convexes.
- (E) Nécessite un calcul de médiane.

45. Quel type d'algorithme nous permet de résoudre le problème des 8 reines le plus efficacement ?

- (A) Un algorithme min max.
- (B) Un algorithme alpha-beta.
- (C) L'algorithme des attrapeurs.
- (D) Un algorithme glouton.
- (E) Un algorithme de retour sur trace.

46. Un algorithme reposant sur une heuristique :

- (A) Donne une valeur approchée du résultat avec un coefficient d'approximation α .
- (B) Ne peut généralement garantir le temps d'exécution et l'optimalité du résultat.
- (C) Ne peut 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.

47. Les positions gagnantes pour J_1 dans un jeu

- (A) Peuvent être calculées en temps polynomial en la taille du jeu.
- (B) Dépendent des réponses de l'adversaire.
- (C) Sont nécessairement dans S_1 , les sommets contrôlés par J_1 .
- (D) Peuvent mener à une défaite de J_1 suivant sa stratégie.
- (E) Aucune des réponses ci-dessus.

3 Concurrency et Synchronisation

48. Un fil d'exécution

- (A) Termine nécessairement.
- (B) Peut partager de la mémoire avec les autres fils d'exécutions.
- (C) A une exécution entrelacée avec les autres fils.
- (D) N'a pas d'utilité si on ne dispose que d'un seul processeur.
- (E) Entraîne un non-déterminisme.

49. L'ordonnanceur est implémenté par le système d'exploitation.

- (A) Vrai
- (B) Faux

50. La création d'une section critique

- (A) Repose sur l'utilisation d'opération atomique.
- (B) Ne peut être assurée sans risque de famine que pour deux fils d'exécution.
- (C) Doit satisfaire le principe d'exclusion mutuelle.
- (D) Sont toujours créés à l'aide de verrous.
- (E) Impose une attente active pour certains fils d'exécution.

51. Parmi les problèmes suivants, lequel est le plus grave ?

- (A) Famine probabiliste
- (B) Famine avérée
- (C) Interblocage
- (D) Situation de compétition
- (E) Ils sont tous aussi graves

52. L'algorithme de la boulangerie de Lamport

- (A) Possède un risque de famine probabiliste, mais pas de famine avérée
- (B) Possède un risque de famine avéré
- (C) Nécessite des opérations de lectures et d'écritures dans le tableau atomiques.
- (D) Nécessite de pouvoir calculer le maximum d'une liste de façon atomique.
- (E) Ne favorise aucun processus.

53. Le mot sémaphore est :

- (A) Masculin
- (B) Féminin

54. Pour les sémaphores, quels noms d'opérations représentent une incrémentation.

- (A) $P(s)$
- (B) $init(s)$
- (C) $lock(s)$
- (D) $wait(s)$
- (E) $post(s)$

4 Logique

55. 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) $((A \vee B) \wedge (C \vee D)) \leftrightarrow (A \wedge C) \vee (B \wedge D)$
- (E) $(A \wedge \neg B) \rightarrow (\neg A \wedge B)$

56. La taille d'une formule est bornée par sa hauteur.

- (A) Vrai
- (B) Faux

57. Une variable libre

- (A) Peut-être liée au même endroit dans la formule.
- (B) Peut-être liée à un autre endroit dans la formule.
- (C) A une portée.
- (D) N'a pas d'incidence sur l'évaluation d'une formule.
- (E) Est toujours associé à une variable liée.

58. Soit φ et ψ deux formules telles que $\psi \models \varphi$. Cela signifie que :

- (A) Que φ est une conséquence de ψ
- (B) Que φ est une sous-alternative à ψ
- (C) Que les valuations satisfaisantes φ satisfont aussi ψ
- (D) Si φ est une antilogie, alors ψ est une antilogie.
- (E) La notation $\models$ n'existe que sous la forme $v \models \varphi$ avec v une valuation.

59. Une équivalence entre formules.

- (A) Signifie qu'elles ont le même arbre de dérivation.
- (B) Peut se prouver par l'étude des tables de vérité.
- (C) Peut se prouver à l'aide de substitution dans des formules qu'on sait équivalentes.
- (D) Ne peut être démontrée sans la déduction naturelle.
- (E) Peut se prouver par une étude de la matrice de confusion.

60. Pour toute formule sans quantificateur, il existe une formule équivalente :

- (A) Sous forme normale conjonctive (D) Sous forme normale de Chomsky
(B) Sous forme normale disjonctive
(C) Sous forme normale négative (E) Avec quantificateur

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

- (A) 2-FNC (D) 3-FND
(B) 3-FNC (E) Avec les opérateurs $\{\top, \perp, \wedge, \vee, \rightarrow\}$, donc pas
(C) 2-FND $\neg$.

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

- (A) Vrai (B) Faux

63. Une règle dérivée

- (A) Peut être utilisée dans une autre preuve. prouvables.
(B) Peut être prouvée par table de vérité.
(C) Permet d'augmenter le nombre de séquents (D) Aucune des réponses ci-dessus.

64. On peut prouver $A \vdash \neg\neg A$ avec les règles de base de la déduction naturelle.

- (A) Vrai (B) Faux

65. En logique propositionnelle, $\Gamma \vdash C$ si et seulement si $\Gamma \models C$

- (A) Vrai (B) Faux

ça dépend du système de règles considéré

5 Théorie des langages

66. Soit $\mathcal{L}$ un langage, alors $\mathcal{L}^*$ est infini.

- (A) Vrai (B) Faux

67. Soit $\mathcal{L}$ et $\mathcal{L}'$ deux langages finis. Alors $\text{Card}(\mathcal{L} \cdot \mathcal{L}') = \text{Card}(\mathcal{L}) \times \text{Card}(\mathcal{L}')$

- (A) Vrai (B) Faux

68. Le langage $\{a^n b^m \mid n \% 2 = m \% 2\}$ est reconnu par l'expression régulière :

- (A) $(aa|bb)^*$ (D) $a^n b^m$
(B) $(aa|ab|ba|bb)^*$
(C) $((a(a|b)^*b)^*|(b(a|b)^*a)^*)^*$ (E) Aucune des réponses ci-dessus.

69. Les langages réguliers sont stables par :

- (A) Préfixe (D) Intersection
(B) Miroir
(C) Complémentaire (E) Différence ensembliste.

70. Un automate fini déterministe complet

- (A) Est nécessairement émondé
- (B) Est unique pour chaque langage (à renommage près)
- (C) Reconnaît l'appartenance d'un mot à son langage en temps linéaire.
- (D) Donne aisément un automate reconnaissant le complémentaire du langage.
- (E) Est toujours de taille exponentielle comparée à un automate non déterministe.

71. La reconnaissance d'un mot par une expression régulière.

- (A) Nécessite au moins une complexité linéaire.
- (B) Peut être effectuée en temps polynomial à partir d'un automate non déterministe.
- (C) Est un problème indécidable.
- (D) N'a pas d'application pratique.
- (E) Aucune des réponses ci-dessus.

72. Parmi les langages suivants, lesquels sont rationnels ?

- (A) $\{a^n b^n \mid n \geq 0\}$
- (B) Les mots bien parenthésés.
- (C) Les mots sur $\{a; b\}$ dont le nombre de a a le même reste dans la division euclidienne par 3
- (D) $\{a^n b^m \mid n \% 2 = m \% 2\}$
- (E) $\{a^p \mid p \text{ premier}\}$

que le nombre de b .

73. Si un langage satisfait le lemme de l'étoile, alors il est rationnel.

- (A) Vrai
- (B) Faux

74. Une dérivation $u \Rightarrow v$ est immédiate si :

- (A) Toutes les variables de u sont dérivées.
- (B) Une seule variable est dérivée et elle se trouve à gauche ou à droite.
- (C) Une seule variable est dérivée.
- (D) v ne contient plus de variables.
- (E) u est l'axiome.

75. Les langages algébriques

- (A) Sont inclus dans les langages rationnels
- (B) Sont stables par étoile de Kleene
- (C) Sont stables par complémentaires
- (D) Sont stables par intersection
- (E) Sont stables par concaténation

76. Quelles sont les grammaires ambiguës ?

- (A) $S \rightarrow SS \mid a$
- (B) $S \rightarrow ST \mid TS \quad T \rightarrow SS$
- (C) $S \rightarrow aSbS \mid bSaS \mid \epsilon$
- (D) $S \rightarrow aSb \mid ab \mid \epsilon$
- (E) $S \rightarrow aS \mid bS \mid \epsilon$

77. Les dérivations droites sont en bijection avec les arbres de dérivations.

- (A) Vrai
- (B) Faux

78. L'analyseur lexical

- ☒ (A) Intervient avant l'analyseur syntaxique
- ☒ (B) Reconnaît des lexèmes
- (C) Permet de représenter un programme sous forme de structure
- ☒ (D) Est construit à partir d'un automate
- (E) Il n'est jamais nécessaire de faire des retours arrière lors de l'analyse lexicale

79. L'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) Interprète la grammaire à considérer en fonction des données d'entrée.
- (E) Aucune des réponses ci-dessus.

80. Savoir si un mot est généré par une grammaire est un problème indécidable.

- (A) Vrai
- ☒ (B) Faux

81. Savoir si une grammaire est ambiguë est un problème indécidable.

- ☒ (A) Vrai
- (B) Faux

6 Complexité

82. Parmi les problèmes ci-dessous, lesquels sont décisionnels ?

- (A) La résolution d'une grille de Sudoku
- ☒ (B) La primalité d'un entier
- (C) Le plus court chemin entre deux points.
- ☒ (D) Le problème de l'arrêt.
- ☒ (E) L'équivalence entre un langage générée par une grammaire et reconnu par un automate.

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

- (A) La résolution d'une grille de Sudoku
- (B) La primalité d'un entier
- ☒ (C) Le plus court chemin entre deux points.
- (D) Le problème de l'arrêt.
- (E) L'équivalence entre un langage générée par une grammaire et reconnu par un automate.

84. Pour quelles complexités peut-on dire qu'un algorithme tourne en temps polynomial ?

- ☒ (A) $O(\log(n!))$
- (B) $2^{o(n)}$
- (C) $O(\log^n(n))$
- ☒ (D) $O(\sqrt{n})$
- (E) $O(2^n)$

85. L'algorithme de primalité cherchant à savoir si n est premier en calculant son modulo par tous les entiers de 2 à $\lceil \sqrt{n} \rceil$ est-il de complexité :

- (A) Logarithmique
- (B) Polynomiale
- (C) Sous-linéaire
- ☒ (D) Exponentielle
- (E) Linéaire

86. La complexité d'un problème correspond à :

- (A) La complexité dans le pire cas du meilleur algorithme y répondant. (C) La complexité dans le meilleur cas du meilleur algorithme y répondant.
- (B) La complexité dans le pire cas du pire algorithme y répondant. (D) Dépend du format d'entrée des instances.
- (E) N'est pas définie.

87. Si un problème A se réduit polynomialement à B alors

- (A) Si A se résout en temps polynomial, alors B aussi. (C) Plus facile que B .
- (B) Si B se résout en temps polynomial, alors A aussi. (D) Une instance de A peut être modélisée par une instance de B .
- (C) Le problème A est considéré comme plus difficile que B . (E) Si A est décidable alors B aussi.

88. Si un problème est dans NP alors

- (A) On ne sait pas le résoudre en temps polynomial. (D) Il s'agit d'un problème fonctionnel.
- (B) Il peut être résolu en temps exponentiel. (E) Pour toute instance de notre problème, on peut créer un certificat en temps polynomial.
- (C) Il peut être résolu en espace polynomial.

89. Un certificat pour le problème k -COLOR est :

- (A) Un graphe k -coloriable. (D) Une k -coloration de notre instance.
- (B) Une partition de notre instance en k stables. (E) Le mot vide.
- (C) Un algorithme de coloration de graphe.

90. La NP-complétude d'un problème dépend de la représentation des instances.

- (A) Vrai (B) Faux

91. 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. (C) Admet une $\frac{3}{2}$ -approximation.
- (B) Est NP-complet dans ses deux versions. (D) N'admet pas de λ -approximation.
- (E) Peut-être résolu en espace polynomial.

92. Sous réserve que $P \neq NP$, les problèmes suivants sont-ils NP-complet ?

- (A) Le problème de l'arrêt. (D) La reconnaissance d'un mot par une expression régulière.
- (B) L'existence d'un chemin eulérien. (E) Le problème du sac à dos.
- (C) L'existence d'un chemin hamiltonien.

93. Un problème d'optimisation P associé à un problème de décision Q

- (A) Ne peut être approximé si Q est NP-complet. (C) Plus difficile que P .
- (B) On ne sait pas le résoudre en temps polynomial si P est NP-complet. (D) Est dans NP si Q est dans NP.
- (C) Peut être résolu en temps polynomial si Q est dans P. (E) Aucune des réponses ci-dessus.

94. Les problèmes indécidables

- (A) Ne sont pas les mêmes suivant le langage de programmation.
- (B) Ne peuvent être réduit à un autre problème.
- (C) Ne concerne que des problèmes se rapportant au fonctionnement d'algorithmes.
- (D) Dépendent de l'implémentation des instances.
- (E) Aucune des réponses ci-dessus.

95. Quels problèmes sont 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 formules propositionnelles.

96. La machine universelle est :

- (A) Un programme capable de simuler l'exécution d'un autre programme.
- (B) Un programme capable de résoudre tous les problèmes informatique.
- (C) Un programme capable de prendre des entrées de n'importe quel type.
- (D) Considère un programme et son entrée comme de même nature.
- (E) Repose significativement sur la possibilité de tout encoder avec un type universel.

7 Quelques aspects techniques

97. Si le type de p est `int**` et est correctement alloué alors les objets suivants sont bien définis :

- (A) `*p`
- (B) `p[1][1]`
- (C) `***p`
- (D) `p**`
- (E) `&&p`

98. La commande suivante `int* p1 = p; free(p); int x = *p1;` est-elle dans la norme ?

- (A) Oui
- (B) Non

99. `int x=2; int* p1=&x; int** p2 = &p1; free(p1); free(p2);` est dans la norme.

- (A) Oui
- (B) Non

100. Si les droits d'un fichier sont `rw-r-xr-` alors

- (A) tout le monde a toutes les permissions
- (B) tout le monde peut écrire sur le fichier.
- (C) ceux qui peuvent lire le fichier peuvent aussi écrire dessus.
- (D) le propriétaire et son groupe sont les seuls à pouvoir écrire sur le fichier.
- (E) aucune des réponses précédentes

FIN