IN 101
3 mai 2010

Rattrapage

ENSTA ParisTech

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] Parmi les fonctions suivantes, laquelle retourne le PGCD de `a` et `b` ?

☐
```
int f(int a, int b) {
  if (b == 0) {
    return a%b;
  }
  return f(b, a);
}
```

■
```
int f(int a, int b) {
  if (b == 0) {
    return a;
  }
  return f(b, a%b);
}
```

☐
```
int f(int a, int b) {
  if (a == 0) {
    return b;
  }
  return f(a, a%b);
}
```

☐
```
int f(int a, int b) {
  if (a = 0) {
    return b;
  }
  return f(a, b%a);
}
```

2] À quelle ligne du code suivant y a-t-il une erreur de syntaxe qui l'empêche de compiler ?

```
1 int g(int a, int b) {
2   int i,j=0;
3   for (i=0, i<a, i++) {
4     j += b;
5   }
6   return(j);
7 }
```

☐ ligne 1, ☐ ligne 2, ■ ligne 3, ☐ ligne 6.

3] Quel temps met un processeur actuel pour exécuter un calcul de 2^{30} additions (il s'agit d'un ordre de grandeur, pas du temps exact) ?

☐ une microseconde,
■ une seconde,
☐ quelques heures,
☐ une semaine.

4] Laquelle des fonctions suivantes ne calcule pas la somme des entiers de 1 à `n` ?

☐
```
int s(int n) {
  int i,j=0;
  for (i=0; i<=n; i++) {
    j = j+i;
  }
  return j;
}
```

■
```
int s(int n) {
  int i=0;
  while (n > 0) {
    i += n;
  }
  return i;
}
```

☐
```
int s(int n) {
  if (n > 0) {
    return n + s(n-1);
  }
  return 0;
}
```

☐
```
int s(int n) {
  return (n*(n+1))/2;
}
```

5] La fonction suivante affiche le contenu de quel type de structure ?

```
void affiche(void* a) {
  if (a != NULL) {
    printf("%d ", a->val);
    affiche(a->next);
  } else {
    printf("\n");
  }
}
```

☐ un tableau, ■ une liste chaînée, ☐ un arbre, ☐ un tas.

6] Quelle est la complexité de la fonction récursive suivante ?

```
int f(unsigned int a, int b) {
  if (a == 0) {
    return b;
  }
  return f(a-1, a*b);
}
```

■ $\Theta(a)$, ☐ $\Theta(a \times b)$, ☐ $\Theta(b^a)$, ☐ $\Theta(a^b)$.

7] Laquelle de ces complexités est la plus grande ?

☐ $\Theta(\log(n!))$, ☐ $\Theta(n^3)$, ■ $\Theta(2^n)$, ☐ $\Theta(n^{\sqrt{n}})$.

8] Partant d'un matrice d'entiers M de taille 3×3, on veut calculer sa puissance M^n. Quelle est la complexité du meilleur algorithme pour ce calcul (on considère que tous les calculs d'entiers se font en temps constant) ?

■ $\Theta(\log n)$, ☐ $\Theta(n \log n)$, ☐ $\Theta(n^3)$, ☐ $\Theta(3^n)$.

9] Quelle est la complexité en moyenne d'un algorithme de tri à bulles sur un tableau de n entiers ?

☐ $\log n$, ☐ n, ☐ $n \log n$, ■ n^2.

10] Lequel des algorithmes de tri suivants est le plus rapide, si on lui donne en entrée un tableau déjà trié ?

☐ le tri fusion, ■ le tri par insertion,
☐ le tri rapide, ☐ le tri par tas.

11] Dans un algorithme de tri rapide, quel est le meilleur choix (du point de vue de la complexité de l'algorithme) pour l'élément pivot (en supposant qu'il soit possible de connaître la position de cet élément en temps constant) ?

☐ le plus grand élément,
■ l'élément médian,
☐ un élément aléatoire,
☐ cela n'a pas d'importance.

12] La complexité *dans le pire cas* d'un tri par comparaison n'est jamais inférieure à $\Theta(n \log n)$. À quelle condition est-ce vrai *en moyenne* aussi ?

■ si on considère toutes les entrées de taille n équidistribuées,
☐ si on trie des entiers uniquement,
☐ si le tri doit être fait "en place",
☐ si plusieurs des éléments à trier peuvent être égaux.

13] Quelle est la complexité asymptotique d'un algorithme récursif de type « diviser pour régner » si pour une donnée de taille n : l'étape de division coûte $\Theta(n)$, cela donne 3 sous-problèmes de taille $\frac{n}{3}$ à résoudre et la fusion des résultats coûte $\Theta(n)$?

☐ $\Theta((\log n)^3)$, ☐ $\Theta(n)$, ■ $\Theta(n \log n)$, ☐ $\Theta(n^3)$.

14] Quelle est la complexité spatiale du stockage de n entiers dans une liste chaînée ?

☐ $\Theta(\log n)$, ■ $\Theta(n)$, ☐ $\Theta(n \log n)$, ☐ $\Theta(n^2)$.

15] Quelle est la complexité de la recherche du i-ème élément d'une liste chaînée de n éléments ?

☐ $\Theta(\log n)$, ■ $\Theta(n)$, ☐ $\Theta(n \log n)$, ☐ $\Theta(n^2)$.

16] Lorsque l'on implémente (de façon standard) une *file* à l'aide d'une liste simplement chaînée, à quelle position de la liste se trouve l'élément le plus ancien (celui qui a été inséré en premier) ?

■ en première position,
☐ en dernière position,
☐ en première ou en dernière position (c'est un choix d'implémentation),
☐ cela dépend de la valeur des éléments insérés.

17] Considérons une *pile* munie des fonctions `push` et `pop` qui servent respectivement à ajouter et extraire un élément de cette pile. On exécute la suite de commandes : `push(3), pop(), push(5), push(7), push(4), pop(), push(6), pop(), pop()`. Quels éléments sont retournés (dans l'ordre) par les appels à la fonction `pop` ?

☐ 3,5,7,4, ■ 3,4,6,7, ☐ 6,4,7,5, ☐ 5,6,4,7.

18] Considérons maintenant une *file* munie elle aussi des fonctions `push` et `pop` qui servent respectivement à ajouter et extraire un élément de cette file. On exécute la suite de commandes : `push(3), pop(), push(5), push(7), push(4), pop(), push(6), pop(), pop()`. À la fin, quel élément reste-t-il dans la file ?

☐ 4,
☐ 5,
■ 6,
☐ cela dépend de l'implémentation.

19] On utilise un algorithme naïf pour rechercher un motif de longueur m dans un texte de n caractères. Cet algorithme teste pour chaque position i si le motif est à la position i du texte. Quel est la complexité moyenne d'un test qui détermine si le motif est effectivement à une position i donnée du texte ?

☐ $\Theta(1)$, ■ $\Theta(m)$, ☐ $\Theta(n)$, ☐ $\Theta(n+m)$.

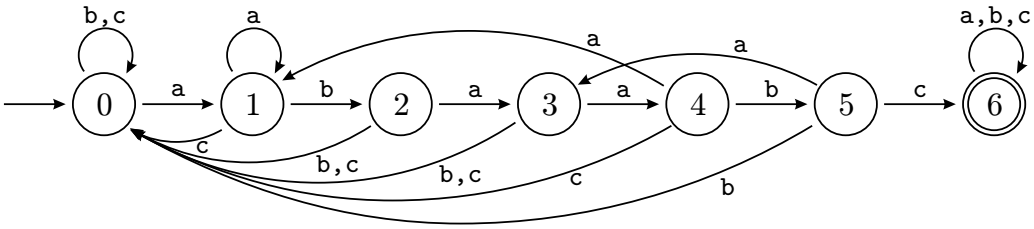


FIGURE 1 – Un automate déterministe.

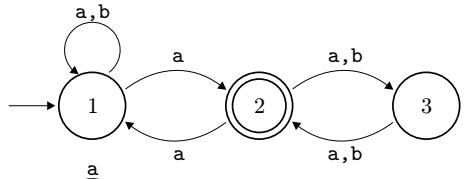
20] L'automate de la FIGURE 1 permet la recherche du motif `abaabc` dans un texte. Cependant il contient une erreur ! De quel état de l'automate la flèche à corriger part-elle ?

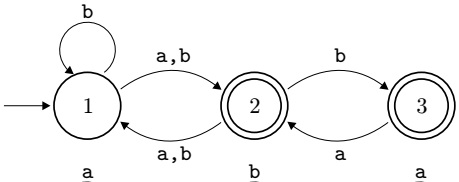
☐ 1, ☐ 2, ■ 3, ☐ 4.

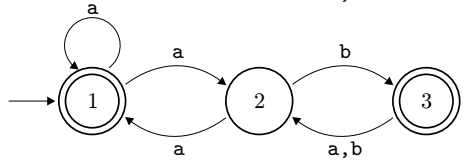
21] Si l'on néglige le temps nécessaire pour construire l'automate, pour lequel des motifs suivants la complexité de sa recherche (à l'aide d'un automate) dans un texte de n caractères sera-t-elle la plus élevée ?

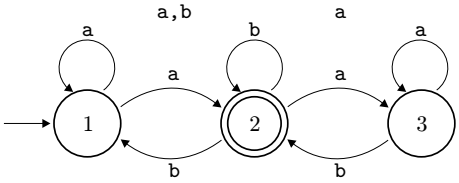
☐ `abaabc`,
☐ `abaaabaab`,
☐ `acdbf`,
■ les trois ont la même complexité.

22] Lequel des automates non-déterministes suivants ne reconnaît pas le motif `abaa` ?

☐


■


☐


☐


23] On implémente un annuaire à l'aide d'une *table à adressage direct* : à un login est associé un ensemble d'informations liées à une personne. Laquelle des opérations suivantes est la plus coûteuse ?

☐ insérer une nouvelle personne,
☐ rechercher une personne ayant un login donné,
☐ supprimer une personne ayant un login donné,
■ ces 3 opérations ont le même coût.

24] Dans l'annuaire de la question précédente, si l'on considère qu'il y a 100 utilisateurs et que les logins sont tous formés de 5 lettres minuscules, quelle est la taille de la table à adressage direct utilisée ?

☐ 100, ■ 26^5,
☐ 26×5, ☐ cela dépend des logins choisis.

25] On décide maintenant d'utiliser une *table de hachage* pour implémenter l'annuaire des 2 questions précédentes. La sortie de la fonction de hachage utilisée dans la table est en entier entre 0 et $k-1$. Toujours pour 100 utilisateurs ayant des logins formés de 5 lettres minuscules, parmi les valeurs suivantes, quel est le meilleur choix pour k ?

■ 100, ☐ 26^5,
☐ 26×5, ☐ cela dépend des logins choisis.

26] Combien de feuilles un arbre binaire de hauteur h contient-il au maximum ?

☐ h, ☐ $2h$, ■ 2^h, ☐ $2^{h+1} - 1$.

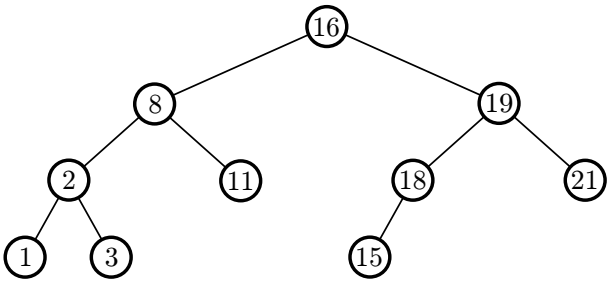


FIGURE 2 – Un arbre binaire.

27] Quels nœuds faut-il échanger pour que l'arbre de la FIGURE 2 vérifie les propriétés d'un *arbre binaire de recherche* ?

- ☐ 16 et 18,
- ■ 16 et 15,
- ☐ 15 et 11,
- ☐ il n'y a rien à changer.

28] Si l'on affiche les nœuds de l'arbre de la FIGURE 2 à l'aide d'un parcours *préfixe* que doit-on voir s'afficher ?

- ☐ 1,2,3,8,11,15,16,18,19,21,
- ☐ 1,2,3,8,11,16,15,18,19,21,
- ■ 16,8,2,1,3,11,19,18,15,21,
- ☐ 1,3,2,11,8,15,18,21,19,16.

29] Si on considère maintenant l'arbre de la FIGURE 2 comme un arbre AVL, à quel nœud est-il déséquilibré ?

- ☐ 8,
- ☐ 19,
- ☐ 18,
- ■ il n'est pas déséquilibré.

30] Quelle est la complexité *en moyenne* d'une insertion dans un arbre binaire de recherche de hauteur h contenant n nœuds ?

- ☐ $\Theta(n)$,
- ☐ $\Theta(n \log n)$,
- ☐ $\Theta(\log h)$,
- ■ $\Theta(h)$.

31] Quelle est la complexité *dans le pire cas* d'une suppression dans un arbre AVL de hauteur h contenant n nœuds ?

- ☐ $\Theta(n)$,
- ☐ $\Theta(n \log n)$,
- ☐ $\Theta(\log h)$,
- ■ $\Theta(h)$.

32] Dans un tas, si le plus grand élément est toujours à la racine, où se trouve toujours le plus petit élément ?

- ☐ tout à gauche de l'arbre,
- ☐ le nœud le plus à droite du dernier niveau,
- ■ dans une feuille,
- ☐ dans le sous-arbre gauche de la racine.

33] On implémente un tas à l'aide d'un tableau (comme vu en cours) et la racine du tas est dans la case 0 du tableau. Dans quelle case se trouve le fils gauche du nœud qui est dans la case i ?

- ☐ $\lfloor \frac{i-1}{2} \rfloor$,
- ☐ $2i$,
- ■ $2i+1$,
- ☐ $2(i+1)$.

34] Pour une implémentation standard de tas avec un tableau, lequel des tableaux suivants ne représente pas un tas bien ordonné ?

- ☐

24	22	20	18	17	16	11

- ■

26	24	13	12	11	16	

- ☐

28	17	27	12	15	21	24

- ☐

19	13	17	11	12	15	

35] Quel est l'avantage de l'algorithme de *tri par tas* par rapport à l'algorithme de *tri rapide* (quicksort) ?

- ☐ une meilleure complexité en moyenne,
- ■ une meilleure complexité dans le pire cas,
- ☐ c'est un tri en place,
- ☐ il n'a aucun avantage.

36] Quelle est la complexité d'un algorithme qui convertit un graphe à S sommets et A arêtes d'une structure en matrice d'adjacence vers une structure en liste de successeurs ?

☐ $\Theta(A)$, ☐ $\Theta(S + A)$, ☐ $\Theta(A \times S)$, ■ $\Theta(S^2)$.

37] On effectue un parcours en profondeur d'un graphe orienté sans cycle dans le but de faire du tri topologique. Lequel des éléments suivants est indispensable pour effectuer correctement le tri topologique à l'issue du parcours ?

- ☐ le tableau des dates de début de visite des nœuds,
- ■ le tableau des dates de fin de visite des nœuds,
- ☐ le tableau des pères,
- ☐ aucun de ces trois tableaux.

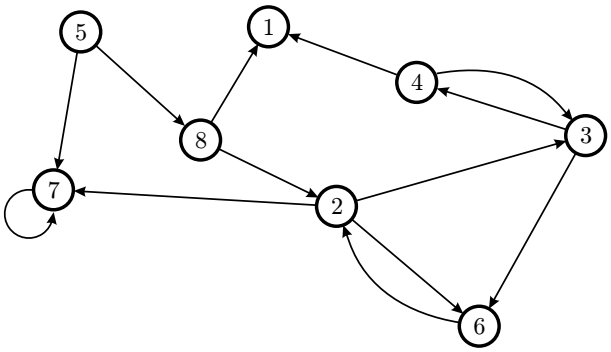


FIGURE 3 – Un graphe orienté.

38] Dans le graphe de la FIGURE 3, quelle est la longueur du plus long chemin sans cycles ?

☐ 3, ☐ 4, ■ 5, ☐ 7.

39] Si on effectue un parcours en profondeur du graphe de la FIGURE 3 en partant du sommet 5, lequel des sommets suivants ne peut en aucun cas être visité en dernier (on considère un nœud comme visité dès le début de sa visite) ?

■ 3, ☐ 4, ☐ 6, ☐ 7.

40] Combien d'arêtes comprend un recouvrement du graphe de la FIGURE 3 partant du sommet 5 ?

■ 7, ☐ 8, ☐ 10, ☐ 13.

41] Si on effectue un parcours en largeur du graphe de la FIGURE 3 en partant du somment 5, quel sera le père du sommet 2 ?

■ 8, ☐ 6, ☐ 3, ☐ 7.

42] L'algorithme de Aho, Hopcroft, Ullman de calcul de plus courts chemins dans un graphe utilise une représentation de graphe par :

- ☐ un tableau de pères,
- ■ une matrice d'adjacence,
- ☐ une liste de successeurs,
- ☐ un arbre général.