IN 102
16 mars 2009

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, la note totale peut être négative !

1] Laquelle de ces complexités n'est pas considérée comme une complexité *polynomiale* pour une entrée de taille n ?

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

2] En considérant que le coût d'une addition ou d'une multiplication est 1, quelle est la complexité du meilleur algorithme pour calculer a^b ?

☐ $\Theta(\log a)$, ☐ $\Theta(\log b)$, ☐ $\Theta(a \log b)$, ☐ $\Theta(b)$.

3] Que mesure la complexité spatiale d'un algorithme ?

☐ Le nombre de lignes de C nécessaire pour l'écrire,
☐ le nombre de pointeurs dont on a besoin pour l'exécuter,
☐ la quantité de mémoire nécessaire à son exécution,
☐ le nombre d'appels à la fonction `malloc`.

4] Quelle est la complexité *dans le pire cas* de l'algorithme de tri fusion (pour trier n éléments) ?

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

5] Quel nom donne-t-on à l'algorithme de tri qui à chaque étape choisit un pivot, et sépare les éléments à trier selon qu'ils sont plus petits ou plus grands que ce pivot ?

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

6] Comment appelle-t-on un algorithme de tri qui permet de trier un tableau d'éléments en n'allouant qu'une quantité de mémoire additionnelle indépendante de la taille de tableau ?

☐ un tri *en place*, ☐ un tri *par paquets*,
☐ un tri *par comparaisons*, ☐ un tri *en liste*.

7] Quelle est la plus petite complexité *en moyenne* que peut avoir un algorithme de tri par comparaisons pour trier des tableaux de n entiers équidistribués ?

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

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

☐ une boucle `for`,
☐ une condition de terminaison,
☐ une étape de division,
☐ une étape de fusion.

9] Un algorithme récursif faisant deux fois appel à lui-même avec une taille $n-1$ pour résoudre le problème de taille n peut avoir une complexité telle que $T(n) = 1 + 2 \times T(n-1)$. Quelle est alors sa complexité totale ?

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

10] À quoi ressemble le plus la structure de donnée suivante ?

```
1 struct S {
2   struct S* p;
3   int v;
4 }
```

☐ un tableau,
☐ une liste chaînée,
☐ une table de hachage,
☐ un arbre.

11] Quel est la complexité *en moyenne* de l'accès au i-ème élément d'une liste chaînée de n éléments ?

☐ $\Theta(1)$, ☐ $\Theta(\log n)$, ☐ $\Theta(n)$, ☐ $\Theta(n \log n)$.

12] Quel est la complexité *en moyenne* de la suppression du premier élément d'une liste chaînée de n éléments ?

☐ $\Theta(1)$, ☐ $\Theta(\log n)$, ☐ $\Theta(n)$, ☐ $\Theta(n \log n)$.

13] On cherche à libérer entièrement la mémoire occupée par une liste chaînée. Laquelle de ces fonctions effectue cette opération correctement ?

☐
```
void free_list (cell* L) {
  if (L != NULL) {
    free_list(L->next) ;
    free(L) ;
  }
}
```

☐
```
void free_list (cell* L) {
  if (L != NULL) {
    free(L) ;
    free_list(L->next) ;
  }
}
```

☐
```
void free_list (cell* L) {
  if (L != NULL) {
    free_list(L) ;
    free(L->next) ;
  }
}
```

☐
```
void free_list (cell* L) {
  if (L != NULL) {
    free(L->next) ;
    free_list(L) ;
  }
}
```

14] Parmi les structures de données suivantes, laquelle peut-on utiliser pour implémenter efficacement une *pile* sans avoir à gérer les problèmes de dépassement de capacité (stack overflow) ?

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

15] Partant d'une *file* vide, on effectue les opérations suivantes : `push(3)`, `push(7)`, `pop()`, `push(17)`, `pop()`, `push(11)`. Quelle valeur devrait alors renvoyer un nouvel appel à `pop()` ?

☐ 3, ☐ 7, ☐ 11, ☐ 17.

16] Quelle est la complexité spatiale d'une table à adressage direct pour référencer une bibliothèque de n livres avec m références possibles au total ?

☐ $\Theta(n)$, ☐ $\Theta(m)$, ☐ $\Theta(n \times m)$, ☐ $\Theta(n + m)$.

17] Dans la bibliothèque précédente, quel est alors le coût de la recherche d'un livre ayant une référence donnée ?

☐ $\Theta(1)$, ☐ $\Theta(n)$, ☐ $\Theta(m)$, ☐ $\Theta(\log n)$.

18] Considérons une table de hachage dont la fonction de hachage prend une clef dans $[0, m-1]$ et retourne un haché dans $[0, k-1]$. Quelle est la complexité *en moyenne* de la recherche d'un élément dans cette table si elle contient n éléments ?

☐ $\Theta(1)$, ☐ $\Theta(k)$, ☐ $\Theta(\frac{n}{k})$, ☐ $\Theta(\log n)$.

19] Dans la table de hachage précédente, quelle est la complexité *en moyenne* de l'insertion d'un nouvel élément ?

☐ $\Theta(1)$, ☐ $\Theta(k)$, ☐ $\Theta(\frac{n}{k})$, ☐ $\Theta(\log n)$.

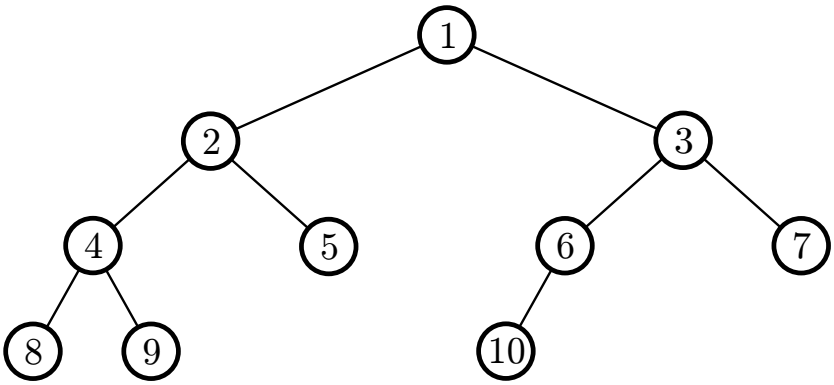

FIG. 1 – Un arbre.

20] Combien de nœuds *internes* l'arbre de la FIG. 1 comporte-t-il ?

☐ 4, ☐ 5, ☐ 7, ☐ 10.

21] Dans quel ordre les nœuds de l'arbre de la FIG. 1 seront ils parcourus lors d'un parcours *préfixe* de l'arbre ?

☐ 1,2,4,8,9,5,3,6,10,7, ☐ 8,4,9,2,5,1,10,6,3,7,
☐ 1,2,3,4,5,6,7,8,9,10, ☐ 8,9,4,5,2,10,6,7,3,1.

22] Dans quel ordre les nœuds de l'arbre de la FIG. 1 seront ils parcourus lors d'un parcours *infixe* de l'arbre ?

☐ 1,2,4,8,9,5,3,6,10,7, ☐ 8,4,9,2,5,1,10,6,3,7,
☐ 1,2,3,4,5,6,7,8,9,10, ☐ 8,9,4,5,2,10,6,7,3,1.

23] Dans quel ordre les nœuds de l'arbre de la FIG. 1 seront ils parcourus lors d'un parcours *en largeur* de l'arbre ?

☐ 1,2,4,8,9,5,3,6,10,7, ☐ 8,4,9,2,5,1,10,6,3,7,
☐ 1,2,3,4,5,6,7,8,9,10, ☐ 8,9,4,5,2,10,6,7,3,1.

24] Par quelle valeur peut-on remplacer le nœud '?' de l'ABR de la FIG. 2 (page suivante) pour que la propriété d'ABR soit respectée ?

☐ 6, ☐ 8, ☐ 10, ☐ 11.

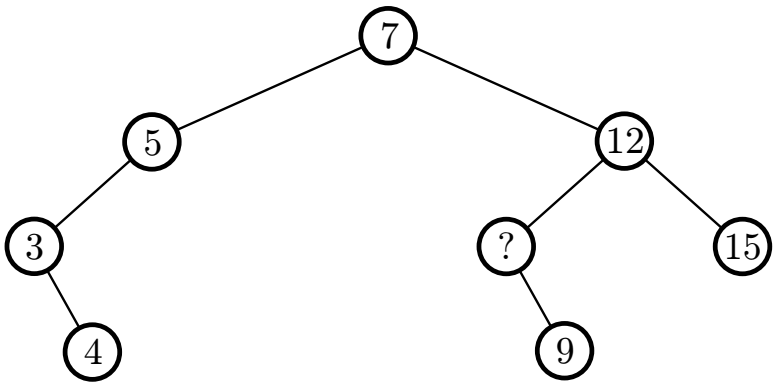

FIG. 2 – Un arbre binaire de recherche.

25] Avec une implémentation standard d'ABR, si l'on supprime successivement les nœuds 5, 3, 7, puis 4 de l'arbre de la FIG. 2, lequel des nœuds restants sera la nouvelle racine de l'arbre ?

☐ 12, ☐ ?, ☐ 9, ☐ 15.

26] Avec une implémentation standard d'ABR, combien de comparaisons de valeurs de nœuds devra-t-on faire pour insérer un nœud portant une valeur de 6 à sa place dans l'ABR de la FIG. 2 ?

☐ 1, ☐ 2, ☐ 3, ☐ 4.

27] Quel est le nombre maximum de fils que peut avoir un nœud juste après son insertion dans un ABR ?

☐ 0, ☐ 1, ☐ 2, ☐ ça dépend.

28] Le plus petit élément d'un tas est *toujours* :

☐ sa racine, ☐ un nœud interne, ☐ une feuille, ☐ aucun des trois.

29] Pour une implémentation standard d'un tas avec un tableau (avec la racine à l'indice 0 du tableau), quel est l'indice du grand-père (le père du père) de l'élément d'indice 12 ?

☐ 0, ☐ 1, ☐ 2, ☐ 3.

30] Quelle est la complexité d'une opération de rotation à gauche à la racine d'un arbre AVL contenant n nœuds ?

☐ $\Theta(1)$, ☐ $\Theta(\log n)$, ☐ $\Theta(n)$, ☐ $\Theta(n \log n)$.

31] Combien de nœuds un arbre AVL de hauteur 3 contient-il *au minimum* ? *(rappelons qu'un arbre contenant un seul nœud est de hauteur 0)*

☐ 4, ☐ 6, ☐ 7, ☐ 15.

32] Si l'on essaye de construire un graphe orienté *sans cycle* à n sommets ayant le plus long chemin possible, quelle sera la longueur maximum de ce chemin ?

☐ $n-1$, ☐ $n(n-1)$,
☐ n, ☐ il n'y a pas de limite.

33] Combien un graphe orienté sans cycles possédant n sommets peut-il posséder d'arcs au maximum ?

☐ $n-1$, ☐ $\frac{n(n-1)}{2}$, ☐ $n(n-1)$, ☐ $2n$.

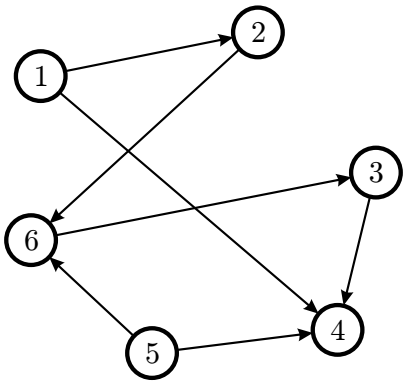

FIG. 3 – Un graphe.

34] Laquelle de ces 4 propriétés *n'est pas vraie* pour le graphe de la FIG. 3 ?

- ☐ c'est un graphe orienté,
- ☐ c'est un graphe sans cycle,
- ☐ c'est un graphe fortement connexe,
- ☐ il n'existe pas d'arbre couvrant pour ce graphe.

35] Si l'on effectue un parcours *en largeur* du graphe de la FIG. 3 en démarrant du sommet 1, quel sommet sera le père du sommet 4 ?

☐ 1, ☐ 3, ☐ 5, ☐ ça dépend.

36] Si l'on effectue un parcours *en profondeur* du graphe de la FIG. 3 en démarrant du sommet 1, quel sommet sera le père du sommet 4 ?

☐ 1, ☐ 3, ☐ 5, ☐ ça dépend.

37] Laquelle de ces matrices est la matrice d'adjacence de la fermeture transitive réflexive du graphe de la FIG. 3 ?

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

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

38] L'algorithme de Aho, Hopcroft, Ullman sert à calculer des plus courts chemins dans un graphe. Quelle est sa complexité pour un graphe à S sommets et A arcs ?

☐ $\Theta(S + A)$, ☐ $\Theta(S \times A)$, ☐ $\Theta(S^2 + A)$, ☐ $\Theta(S^3)$.

39] À quoi sert le modulo dans l'algorithme de Rabin-Karp avec modulo pour la recherche de motifs ?

- ☐ à éviter des biais statistiques,
- ☐ à manipuler des entiers plus petits,
- ☐ à diminuer le nombre de mauvais motifs reconnus,
- ☐ à réduire la longueur du motif recherché.

40] Combien d'états doit avoir un automate déterministe pour la recherche du motif *automate* dans un texte ?

☐ 7, ☐ 8, ☐ 9, ☐ 10.

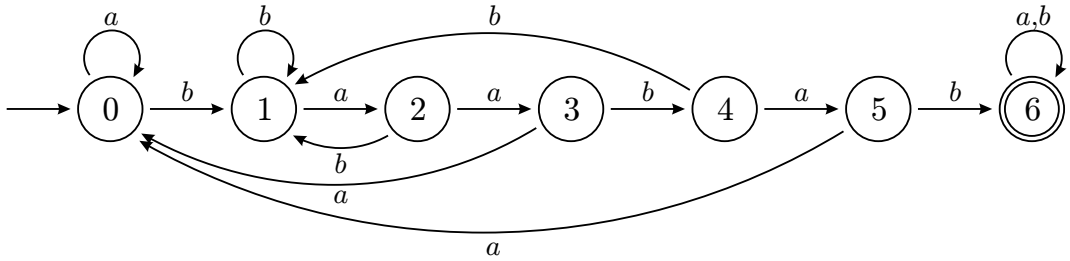

FIG. 4 – Un automate.

41] L'automate de la FIG. 4 devrait reconnaître le motifs *baabab*, mais l'une de ses transitions est fausse. De quel état part cette mauvaise transition ?

☐ 2, ☐ 3, ☐ 4, ☐ 5.

42] Parmi les automates déterministes suivants, lequel reconnaît tous les mots formés avec l'alphabet $\Sigma = \{a, b\}$ et donc le langage $\mathcal{L} = \Sigma^*$?