



QCM de rentrée

1 Pour chacune des affirmations suivantes, dire si elle est vraie ou fausse:

1. $n \in O(n^2)$?
2. $n^2 \in O(n)$?
3. $n \in \Theta(n^2)$?
4. $\log n \in O(n)$?
5. $\log n \in \Theta(n)$?

- 1
- 2
- 3
- 4
- 5

2 Quelle est la taille de la représentation binaire d'un entier n ?

Pour x réel positif, on notera $\text{Ent}(x)$ la partie entière par défaut de x , i.e. le plus grand entier inférieur ou égal à x .

- n
- $n/2$
- $\text{Ent}(\log_2(n))$
- $\text{Ent}(\log_2(n+1))$
- $\text{Ent}(\log_2(n))+1$
- Sans réponse

3 Qu'appelle-t-on le coût d'un algorithme ?

- Le nombre d'opérations arithmétiques (additions, soustractions, ...) élémentaires effectuées lorsqu'on l'exécute.
- Le nombre de comparaisons d'éléments effectuées lorsqu'on l'exécute.
- La place mémoire nécessaire pour l'exécuter.
- Le temps effectif nécessaire à son exécution.
- Toutes les propositions ci-dessus peuvent être correctes selon les cas.
- Aucune proposition ci-dessus n'est correcte.
- Sans réponse

4 Pour calculer le coût en moyenne d'un algorithme, il faut :

- calculer le coût en meilleur cas et le coût en pire cas et faire la moyenne ;
- appliquer plusieurs fois successivement l'algorithme et faire la moyenne des coûts exacts obtenus ;
- Toutes les propositions ci-dessus sont correctes ;
- Aucune proposition ci-dessus n'est correcte
- Sans réponse

5

Soient deux algorithmes A et B. La complexité dans le meilleur des cas comme dans le pire est $\theta(n^2)$. Celle de B est $\theta(n \log n)$.

Que peut-on affirmer:

- B est toujours plus rapide que A.
- A est toujours plus rapide que B.
- B est toujours plus rapide que A pour les données de taille supérieure à une certaine taille.
- A est toujours plus rapide que B pour les données de taille supérieure à une certaine taille.
- Sans réponse

6

Soient les fonctions $f_1(n) = 2^n$, $f_2(n) = n^3$, $f_3(n) = n^{1/2}$, $f_4(n) = \log n$.

Quel est l'ordre de grandeur de ces fonctions suivantes du plus petit au plus grand: $f <= g$ si $f = O(g)$.

- f_3, f_4, f_2, f_1
- f_4, f_3, f_1, f_2
- f_4, f_3, f_2, f_1
- f_1, f_2, f_3, f_4
- Sans réponse

7

Un algorithme prend 1 seconde sur votre machine pour traiter un tableau de 1000 éléments. On notera n le nombre d'éléments.

Combien de temps prendra-t-il pour traiter un tableau de 100.000 éléments si vous supposez que son temps d'exécution est proportionnel à n?

- 2 secondes
- 10 secondes
- entre 1 et 2 minutes
- plus d'un quart d'heure
- Sans réponse

8

Un algorithme prend 1 seconde sur votre machine pour traiter un tableau de 1000 éléments. On notera n le nombre d'éléments.

Combien de temps prendra-t-il pour traiter un tableau de 100.000 éléments si vous supposez que son temps d'exécution est proportionnel à n^2 ?

- quelques secondes
- environ une minute
- quelques minutes
- plus de 2 heures
- Sans réponse

9

Un algorithme prend 1 seconde sur votre machine pour traiter un tableau de 1000 éléments. On notera n le nombre d'éléments.

Combien de temps prendra-t-il pour traiter un tableau de 100.000 éléments si vous supposez que son temps d'exécution est proportionnel à n^3 ?

- moins d'une minute
- moins d'une heure
- moins d'une journée
- moins d'une semaine
- plus d'une semaine
- Sans réponse

10

Un algorithme prend 1 seconde sur votre machine pour traiter un tableau de 1000 éléments. On notera n le nombre d'éléments.

Combien de temps prendra-t-il pour traiter un tableau de 100.000 éléments si vous supposez que son temps d'exécution est proportionnel à $n \log n$?

- moins d'une minute
- moins de deux minutes
- moins de trois minutes
- plus d'une heure
- Sans réponse

11

Quelle est la complexité de:

```
int fun(int n) {  
    int count = 0;  
    for (int i = n; i > 0; i --)  
        for (int j = 0; j < i; j++) count +=1;  
    return count;  
}
```

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

- $\Theta(n^3)$
- Sans réponse

12 Quelle est la complexité de:

```
int fun(int n) {  
    int count = 0;  
    for (int i = n; i > 0; i /= 2)  
        for (int j = 0; j < n; j++) count ++;  
    return count;  
}
```

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

13 Quelle est la complexité de:

```
void myst (int n; int arr[]) {  
    int i, j = 0;  
    for (i= 0; i < n; ++i)  
        while ((j < n) && (arr[i] < arr[j])) j++;  
}
```

- $\theta(\log n)$
- $\theta(n)$
- $\theta(n \log n)$
- $\theta(n^2)$
- Sans réponse

14 Pour lequel de ces deux problèmes existe-il un algorithme?

- Entrée: G une grammaire algébrique; Sortie: Oui Ssi le langage généré est non vide
- Entrée: G une grammaire algébrique; Sortie: Oui Ssi le langage généré est l'ensemble de tous les mots sur l'alphabet.
- Sans réponse

Fermer cette fenêtre