

Contrôle : Big Data

28 Septembre 2020

Durée 1h, tous documents non électroniques autorisés.

Le barème est donné à titre indicatif.

Dans tous les exercices sur la complexité, nous supposons, sauf indication contraire, que les opérations suivantes prennent un temps constant $O(1)$: addition de deux nombres, multiplication de deux nombres, division de deux nombres, soustraction de deux nombres et la racine carrée.

1. (2 points) Quels sont les risques associés au stockage et traitement de données volumineuses ?
2. (2 points) Donnez au moins trois exemples du matériel informatique pouvant être utilisé pour le traitement de grandes quantités de données.

Nous voulons calculer la valeur 2^n . Voici l'algorithme naïf.

```
def puissance_de_2 (n) :  
    p = 1  
    i = 0  
    while (i < n) :  
        p = p * 2  
        i = i + 1  
    return p
```

3. (2 points) Combien de multiplications fait cet algorithme pour calculer 2^n ?
4. (4 points) Écrire un algorithme qui calcule 2^n en utilisant moins de multiplications que l'algorithme naïf.
5. (2 points) Combien de multiplications fait votre algorithme pour calculer 2^n ?
6. (3 points) Quel algorithme préféreriez-vous pour faire un traitement de grandes quantités de données ? Pourquoi ?
 - A. un algorithme avec la complexité $O(n \log_2(n))$;
 - B. un algorithme avec la complexité $O(2^{\log_3(n)})$;
 - C. un algorithme avec la complexité $O(n^2)$;
 - D. un algorithme avec la complexité $O(n)$;
 - E. un algorithme avec la complexité $O(2^{\log_2(n)})$.

Choisissons et fixons un algorithme **COMP** de compression de données. Considérons l'ensemble des chaînes binaires de longueur n . Une chaîne binaire de longueur n est dite *compressible* si elle peut être représentée, en utilisant l'algorithme **COMP**, comme une chaîne de longueur strictement inférieure à n .

7. (5 points) Prouvez qu'il existe des chaînes non compressibles, quelque soit l'algorithme **COMP**.