

Interrogation  
 Durée : 1 heure 30 minutes <11 h 20-12 h 50>

**Exercice 1 (complexité du tri) :**

On considère l'algorithme suivant de tri par insertion d'un tableau A de n entiers :

**Entrée :** Un tableau A de n entiers  
**Sortie :** Le tableau A trié par ordre croissant

```

TRI-INSERTION(A){
  Pour j=2 à n{
    clé= A[j]
    i=j-1
    Tant que i>0 et A[i]>clé{
      A[i+1]=A[i]
      i=i-1
    }
    A[i+1]=clé
  }
}
  
```

1. Calculez en fonction de n le nombre  $T(n)$  d'opérations dans le pire des cas de l'algorithme. Expliquez.
2. Trouvez une fonction  $f(n)=n^k$  (k constante) vérifiant  $T(n)=O(f(n))$  et  $f(n)=O(T(n))$ . Expliquez.
3. Que pouvez-vous déduire de la réponse à la question 2 ?

**Exercice 2 (NP-complétude) :**

1. Illustrez à travers un exemple la notion d'instance d'un problème.
2. Définissez les notions suivantes de la théorie de la NP-complétude :
  - a. Certificat
  - b. Algorithme de validation
  - c. La classe NP
  - d. Problème NP-complet
3. Donnez un algorithme polynomial de validation pour le problème SAT (SATisfiabilité). Utilisez la terminologie vue en TP, en TD et en cours. Expliquez.

**Exercice 3 (Structures de données) :** Une pile est une structure de données mettant en œuvre le principe « dernier entré premier sorti » (LIFO : Last In First Out). On considère ici le cas d'une pile implémentée avec un tableau.

1. Une pile doit être initialisée. Expliquez comment.
2. Ecrivez les différentes fonctions et procédures permettant la gestion d'une pile.