

Documents interdits, à l'exception d'une feuille A4 à rendre avec votre copie.

La notation tiendra compte de la présentation et de la clarté de la rédaction.

Le barème, approximatif, est donné sur 10 points puisque l'épreuve dure 1h ($\frac{10}{20} = 0,5 = \frac{1h}{2h}$).

★ **Questions de cours.** (2pt)

▷ **Question 1:** (1pt) À quelles classes de complexité (en notation Θ) appartiennent les algorithmes 1 et 2 suivants ?

```

1  _____ algorithme 1 _____
2  pour i = 1 à n faire
3  pour j = 1 à n faire
   x += 3
  
```

```

1  _____ algorithme 2 _____
2  pour i = 1 à n faire
3  pour j = 1 à n faire
   x += 3
4  pour i = 1 à n faire
5  y = x + 5
  
```

▷ **Question 2:** (1pt) Définissez les types de récursivité suivants : terminale, générative, mutuelle (ou croisée) et structurale.

★ **Exercice 1: Code récursif mystère** (5pts). Considérez le code mystère suivant.

▷ **Question 1:** ($\frac{1}{2}$ pt) Explicitez les appels récursifs effectués pour `puzzle(25,4)`.

▷ **Question 2:** (1pt) Calculez le résultat de la fonction `puzzle` pour les valeurs suivantes. (10,1) (10,2) (10,3) (10,4) (10,6) (10,8) (10,10) Que semble calculer `puzzle()` ?

▷ **Question 3:** ($\frac{1}{2}$ pt) Montrez la terminaison de cet algorithme.

▷ **Question 4:** ($\frac{1}{2}$ pt) Quelle est la complexité algorithmique de `puzzle` (en nombre d'appels récursifs) ?

▷ **Question 5:** ($\frac{1}{2}$ pt) Est-il possible de dérécurser directement cette fonction ? Pourquoi ?

▷ **Question 6:** (2pts) Dérécursez cette fonction en appliquant les méthodes vues en cours (en une ou plusieurs étapes). Explicitez ce que vous faites et pourquoi.

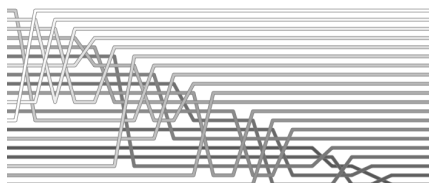
```

1  public int puzzle(int i, int j) {
2      if (i == 1)
3          return j;
4      if (i % 2 == 1)
5          return j+puzzle(i/2,j*2);
6      else
7          return puzzle(i/2,j*2);
8  }
  
```

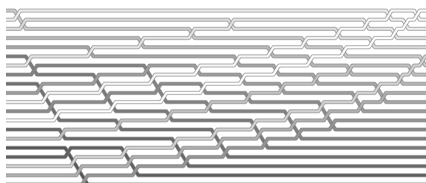
★ **Exercice 2: Identification d'algorithmes de tri** (d'après Aldo Cortesi – 3pt).

Les schémas suivants montrent le fonctionnement de divers algorithmes de tris. Chaque trait grisé indique une valeur, et l'axe des abscisses montre le temps qui passe tandis que l'axe des ordonnées montre la position de chaque valeur (=trait) dans les différentes cases du tableau. La case n°1 est en haut, et la case n°20 est en bas, et une couleur plus claire signifie une valeur plus petite. Quand deux traits se croisent, c'est que l'algorithme a inversé les deux valeurs à cet instant.

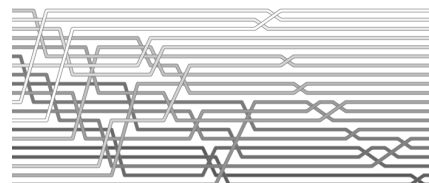
▷ **Question 1:** Identifiez le comportement des algorithmes suivants : tri à bulle, tri par insertion, tri par sélection, shell sort, quick sort. Argumentez vos réponses.



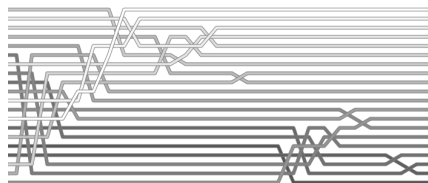
Algorithme A.



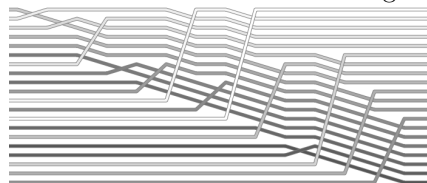
Algorithme B.



Algorithme C.



Algorithme D.



Algorithme E.