

## Algorithmique — Complexité et algorithmes fondamentaux

Nom : \_\_\_\_\_

Classe : \_\_\_\_\_

Au contrôle, une réponse juste sans calcul ne rapporte presque rien : chaque problème se rédige en quatre temps — **justifier**, **poser l'égalité**, **mener le calcul** jusqu'au bout, **conclure** par une phrase avec l'unité. La place réservée est dimensionnée pour ces quatre temps.

**1** Vocabulaire. Répondre en une phrase précise.

- Que mesure la complexité ? Est-ce une durée ?
- Ranger par coût croissant :  $O(n^2)$ ,  $O(1)$ ,  $O(n \log n)$ ,  $O(n)$ ,  $O(\log n)$ .
- Quelle condition la recherche dichotomique exige-t-elle ?
- Quelle est la complexité du tri fusion ? Du tri par insertion dans le cas défavorable ?
- Que calcule l'algorithme de Dijkstra ? Sous quelle condition ?

[illegible]

**2 Comparer les ordres de grandeur. On raisonne en nombre d'opérations.**

- Combien d'opérations pour un algorithme en  $O(n^2)$  avec  $n = 1\,000$  ?
- Combien d'étapes pour une dichotomie sur  $1\,000$  éléments ?
- Calculer le rapport entre les deux.
- Si  $n$  passe à  $2\,000$ , par combien le coût quadratique est-il multiplié ?
- Et le coût de la dichotomie ?

[illegible]

**3 Un développeur applique une recherche dichotomique à un tableau non trié. Analyser.**



- a. Le programme va-t-il planter ?
- b. Que va-t-il renvoyer pour un élément pourtant présent ?
- c. Pourquoi cette faute est-elle plus dangereuse qu'une erreur qui interrompt le programme ?
- d. Faut-il trier avant de chercher ? Dans quel cas est-ce rentable ?

---

---

---

---

---

**4 Choisir un algorithme. Raisonner.**



- a. Pourquoi comparer deux algorithmes sur un petit jeu de données est-il trompeur ?
- b. Dans quel cas le tri par insertion est-il préférable au tri fusion ?
- c. Quel coût le tri fusion a-t-il que le tri par insertion n'a pas ?
- d. Pourquoi Dijkstra échoue-t-il avec des poids négatifs ?
- e. Que signifie qu'il n'existe pas de « meilleur algorithme » dans l'absolu ?

---

---

---

---

---

---

---

---

---

---

Corrigé détaillé sur [progressschool.fr](https://progressschool.fr) — chapitre « Algorithmique — Complexité et algorithmes fondamentaux ».