



SPÉ NSI · TERMINALE

✓ EXIGIBLE AU BAC

PRIORITÉ P1

Diviser pour régner

principe et application au tri fusion

🎯 À LA FIN DE CETTE FICHE, TU SAURAS...

- ✓ Décrire les trois étapes du principe diviser pour régner.
- ✓ Dérouler les étapes du tri fusion sur une liste d'exemple.
- ✓ Déterminer la complexité en temps et en espace du tri fusion.

TON PARCOURS DE RÉVISION — coche chaque partie une fois révisée !

- | | |
|--|---|
| ☐ 1 · Les mots à connaître | ☐ 7 · La carte mentale |
| ☐ 2 · Les règles essentielles | ☐ 8 · Trouve l'erreur |
| ☐ 3 · Méthode | ☐ 9 · Exercices type Bac |
| ☐ 4 · Exemple type Bac | ☐ 10 · QCM chrono (5 min) |
| ☐ 5 · Les pièges à éviter | ☐ 11 · Ta carte « 30 secondes » |
| ☐ 6 · Tips pour t'en souvenir | ☐ ∞ Corrections (pour les parents) |

👤 LE COIN DES PARENTS — pas besoin de connaître le cours, tout est justifié.

- Aider à répartir les révisions de ce chapitre sur plusieurs jours (planning de répétition espacée J+1/J+3/J+7/J+14), sans se prononcer sur le contenu.
- Consulter avec l'élève le calendrier des échéances (DS, épreuves de spécialité, Grand Oral) et vérifier qu'aucune date n'est oubliée.
- Observer les signaux d'alerte (fatigue persistante, stress, décrochage) et en parler avec l'élève, sans se substituer aux professionnels du lycée.

1 Les mots à connaître *surligne ceux que tu confonds !*

Diviser pour régner — la stratégie algorithmique qui résout un problème en le divisant en sous-problèmes plus petits de même nature, en résolvant chaque sous-problème séparément, puis en combinant leurs résultats pour obtenir la solution du problème initial.

Sous-problème — une instance plus petite du même problème que le problème initial, obtenue en divisant les données, et résolue par le même algorithme appliqué récursivement.

Tri fusion — un algorithme de tri qui applique le principe diviser pour régner : il divise la liste à trier en deux moitiés, trie récursivement chaque moitié, puis fusionne les deux moitiés triées en une seule liste triée.

Fusion — l'opération qui combine deux listes déjà triées en une seule liste triée, en comparant à chaque étape leurs premiers éléments restants et en plaçant le plus petit dans le résultat.

Récursivité — le principe de programmation dans lequel une fonction résout un problème en s'appelant elle-même sur une version plus petite ou plus simple de ce même problème.

Condition d'arrêt — le test, présent dans une fonction récursive, qui permet de stopper les appels successifs à elle-même lorsqu'un cas simple est atteint, sans quoi la fonction s'appellerait indéfiniment.

Complexité algorithmique — une mesure du coût d'exécution d'un algorithme en fonction de la taille n des données traitées, exprimée le plus souvent en nombre d'opérations élémentaires (complexité en temps) ou en quantité de mémoire utilisée (complexité en espace).

Notation O (grand O) — la notation qui exprime une complexité algorithmique dans le pire des cas : $O(1)$ signifie un coût constant, indépendant de n ; $O(n)$ signifie un coût proportionnel à n .

Niveau de récursion — chacune des étapes successives de division par deux effectuées par le tri fusion pour passer d'une liste de taille n à des listes d'un seul élément. Leur nombre correspond, en notation mathématique, à $\log n$: le nombre de fois qu'on peut diviser n par deux avant d'atteindre 1.

Cas de base — le cas le plus simple d'un problème, résolu directement sans nouvel appel récursif ; c'est ce cas que la condition d'arrêt est chargée de détecter.

2 Les règles essentielles *chaque règle avec son exemple*

- 1 **Le principe diviser pour régner résout un problème en trois étapes : DIVISER le problème en sous-problèmes plus petits de même nature, RÉGNER en résolvant chaque sous-problème récursivement jusqu'au cas de base. Puis COMBINER leurs résultats pour reconstruire la solution du problème initial.**

exemple → Le tri fusion illustre ces trois étapes sur une liste : il la divise en deux moitiés, trie récursivement chacune, puis fusionne les deux moitiés triées.

- 2 **Le tri fusion divise la liste à trier en deux moitiés à chaque appel récursif, jusqu'à atteindre des listes d'un seul élément, déjà triées par définition : cette étape de division correspond au cas de base de la récursivité.**

exemple → Une liste de 8 éléments est divisée en deux listes de 4, puis chacune en deux listes de 2, puis chacune en deux listes de 1 élément.

- 3 **La fusion combine deux listes déjà triées en comparant à chaque étape leurs premiers éléments restants et en plaçant le plus petit dans la liste résultat, jusqu'à épuiser les deux listes.**

exemple → Fusionner [2, 5] et [1, 3] donne d'abord 1 (plus petit que 2), puis 2, puis 3, puis 5, soit [1, 2, 3, 5].

- 4 **Le tri fusion a une complexité en temps $O(n \log n)$. La division de la liste crée un nombre de niveaux de récursion égal à $\log n$ (chaque niveau divise la taille par deux), et chaque niveau effectue au total n comparaisons de fusion.**

exemple → Pour une liste de 8 éléments, la division crée 3 niveaux de récursion (8 puis 4 puis 2 puis 1), et chaque niveau fusionne au total 8 éléments.

- 5 Le tri fusion utilise une mémoire supplémentaire proportionnelle à n pour stocker les listes intermédiaires créées par la fusion, ce qui lui donne une complexité en espace $O(n)$.**

exemple → Fusionner deux listes de 4 éléments chacune nécessite une liste temporaire de 8 éléments pour stocker le résultat.

- 6 Un sous-problème doit être une instance plus petite du même problème pour que le principe diviser pour régner s'applique : trier une moitié de liste est bien un problème de tri, en plus petit.**

exemple → Trier la moitié gauche d'une liste de 8 éléments revient à trier une liste de 4 éléments, exactement le même type de problème.

- 7 Dès que l'une des deux listes est épuisée pendant la fusion, tous les éléments restants de l'autre liste sont recopiés directement à la suite, dans leur ordre. Ils sont déjà triés et tous plus grands que tous les éléments déjà placés dans le résultat.**

exemple → En fusionnant $[2, 5]$ et $[1, 3]$, une fois 1, 2 et 3 placés, la liste $[1, 3]$ est épuisée : le seul élément restant de l'autre liste, 5, est recopié directement, sans nouvelle comparaison.

3 Méthode — Dérouler le tri fusion sur une liste

Exemple suivi : la liste $[5, 2, 8, 1]$.

Étape 1 Diviser la liste en deux moitiés : ici $[5, 2]$ et $[8, 1]$.

Étape 2 Diviser récursivement chaque moitié jusqu'à des listes d'un seul élément : $[5], [2], [8], [1]$ (cas de base, déjà triées).

Étape 3 Fusionner les listes d'un élément deux à deux : $[5]$ et $[2]$ fusionnent en $[2, 5]$; $[8]$ et $[1]$ fusionnent en $[1, 8]$.

Étape 4 Fusionner les deux listes triées obtenues : $[2, 5]$ et $[1, 8]$ fusionnent en comparant leurs premiers éléments restants (1 avant 2, puis 2 avant 8, puis 5 avant 8). La liste $[2, 5]$ est alors épuisée : le seul élément restant de l'autre liste, 8, est recopié directement, ce qui donne $[1, 2, 5, 8]$.

4 Exemple type Bac — rédigé pas à pas

Déroule le tri fusion sur la liste $[7, 3, 6, 2]$ et donne le résultat final trié.

Étape 1 — Diviser la liste en deux moitiés : $[7, 3]$ et $[6, 2]$.

Étape 2 — Diviser chaque moitié en listes d'un élément : $[7], [3], [6], [2]$ (cas de base).

Étape 3 — Fusionner deux à deux : $[7]$ et $[3]$ donnent $[3, 7]$; $[6]$ et $[2]$ donnent $[2, 6]$.

Étape 4 — Fusionner $[3, 7]$ et $[2, 6]$: comparer 3 et 2 (2 plus petit, il sort en premier), puis 3 et 6 (3 plus petit), puis 7 et 6 (6 plus petit). La liste $[2, 6]$ est alors épuisée : son seul élément restant, 7, est recopié directement.

Conclusion : La liste triée obtenue est $[2, 3, 6, 7]$.

5 Les pièges à éviter *les erreurs classiques en DS...*

X Confondre « diviser » et « trier directement » : le tri fusion ne trie pas en divisant, il trie en FUSIONNANT des sous-listes déjà triées ; la division sert seulement à obtenir des listes assez petites pour être triées trivialement.

X Oublier que le cas de base du tri fusion est une liste d'un seul élément (ou vide), toujours considérée comme déjà triée, sans quoi la récursivité ne s'arrête jamais.

X Croire que la complexité $O(n \log n)$ du tri fusion vient d'une seule boucle : elle vient du produit entre le nombre de niveaux de récursion et le nombre d'opérations de fusion à chaque niveau.

X Fusionner deux listes en recopiant d'abord l'une puis l'autre, sans comparer leurs éléments un à un dans l'ordre : cela ne produit pas une liste triée.

6 Tips — pour t'en souvenir

DIVISER pour régner : divise le problème, résous chaque petit morceau, puis combine les résultats — jamais l'inverse.

Le tri fusion NE TRIE PAS en divisant : la division sert juste à obtenir des listes triviales à trier (1 élément), tout le tri se joue dans la FUSION.

$O(n \log n)$: niveaux de récursion ($\log n$, le nombre de divisions par deux) multipliés par les opérations de fusion par niveau (n).

7 La carte mentale *dessiner, c'est mémoriser !*

« DIVISER POUR RÉGNER » au centre, 3 branches colorées : LES 3 ÉTAPES (diviser/régner/combiner), TRI FUSION (division jusqu'au cas de base, puis fusion), COMPLEXITÉ ($O(n \log n)$ temps, $O(n)$ espace). Un exemple par branche. Redessine-la de mémoire après avoir lu la fiche.



ta carte mentale, de mémoire...

8 À toi de jouer — trouve l'erreur

Un élève affirme, en fusionnant les listes triées $[1, 4]$ et $[2, 3]$: « Je recopie d'abord entièrement la première liste, puis la seconde : cela donne $[1, 4, 2, 3]$, qui est le résultat de la fusion. »

→ *cherche l'erreur avant de lire la réponse !*

L'erreur : La fusion ne recopie pas les listes l'une après l'autre : elle compare leurs premiers éléments restants à CHAQUE étape et place le plus petit dans le résultat. Le résultat correct de la fusion de $[1, 4]$ et $[2, 3]$ est $[1, 2, 3, 4]$, pas $[1, 4, 2, 3]$ qui n'est même pas trié. Pour fusionner correctement, il faut comparer 1 et 2 (1 plus petit, il sort en premier), puis 4 et 2 (2 plus petit), puis 4 et 3 (3 plus petit). La liste $[2, 3]$ est alors épuisée : son seul élément restant, 4, est recopié directement. Le résultat $[1, 2, 3, 4]$ est bien trié, contrairement à la simple concaténation $[1, 4, 2, 3]$.

9 Exercices type Bac *corrections en fin de fiche ✂*

Exercice 1. Déroule le tri fusion sur la liste $[9, 1, 5, 3]$: donne les sous-listes obtenues après division complète, puis le résultat de chaque fusion jusqu'à la liste triée finale.

Exercice 2. Fusionne les deux listes déjà triées $[2, 6, 9]$ et $[1, 4]$ en une seule liste triée, en détaillant chaque comparaison.

10 QCM — 10 questions

⌚ 5 MIN MAX • OBJECTIF 8/10

Une seule bonne réponse par question, sans calculatrice — entraînement épreuve d'automatismes du Bac 2026.

1. Le principe diviser pour régner résout un problème en trois étapes, dans quel ordre ?

- ☐ A Diviser, régner, combiner
- ☐ B Combiner, diviser, régner
- ☐ C Régner, combiner, diviser

2. Dans le tri fusion, le cas de base de la récursivité est atteint quand :

- ☐ A La liste contient un seul élément (ou est vide)
- ☐ B La liste contient exactement deux éléments
- ☐ C Toutes les fusions sont terminées

3. La fusion de deux listes déjà triées consiste à :

- ☐ A Recopier la première liste puis la seconde à la suite
- ☐ B Comparer à chaque étape les premiers éléments restants des deux listes et placer le plus petit dans le résultat
- ☐ C Trier chaque liste séparément avant de les recopier

4. Le tri fusion a pour complexité en temps :

- ☐ A $O(n)$
- ☐ B $O(n \times n)$
- ☐ C $O(n \log n)$

5. Pour une liste de 16 éléments, combien de niveaux de récursion le tri fusion crée-t-il avant d'atteindre des listes d'un seul élément ?

- ☐ A 4
☐ B 8
☐ C 16

6. Pourquoi la division du tri fusion s'arrête-t-elle nécessairement ?

- ☐ A Parce qu'un compteur externe limite le nombre d'appels
☐ B Parce que chaque division réduit la taille de la liste de moitié, jusqu'à atteindre le cas de base (liste d'un élément)
☐ C Parce que la fusion empêche toute nouvelle division

7. Un sous-problème du tri fusion, au sens du principe diviser pour régner, est :

- ☐ A Trier une moitié de la liste initiale, un problème de tri en plus petit
☐ B Une opération de fusion entre deux listes
☐ C Le résultat final trié de la liste complète

8. La complexité en espace du tri fusion, liée aux listes temporaires créées par la fusion, est :

- ☐ A $O(1)$
☐ B $O(n)$
☐ C $O(n \times n)$

9. Pour fusionner les listes triées [3, 8] et [1, 5], la première comparaison à effectuer est :

- ☐ A 3 et 8
☐ B 8 et 5
☐ C 3 et 1

10. Le principe diviser pour régner exige que chaque sous-problème soit :

- ☐ A Toujours résolu par une boucle, jamais par un nouvel appel récursif
☐ B Un problème totalement différent, choisi librement
☐ C Une instance plus petite du même problème que le problème initial



11

En 30 secondes — ta carte de dernière révision

découpe-la et garde-la dans ta trousse — révision de veille de contrôle !

- Diviser pour régner : DIVISER en sous-problèmes, RÉGNER en les résolvant récursivement, COMBINER les résultats.
- Le tri fusion divise la liste en deux jusqu'à des listes d'un élément (cas de base), puis fusionne les listes triées deux à deux.
- Complexité du tri fusion : $O(n \log n)$ en temps, $O(n)$ en espace (listes temporaires de la fusion).

PLAN DE RÉVISION ESPACÉE — relis la carte et refais le QCM : ☐ J+1 ☐ J+3 ☐ J+7 ☐ J+14



QCM en ligne

VA PLUS LOIN — TESTE-TOI SUR CETTE NOTION

📱 Scanne le QR code (ou clique dessus) : le QCM chronométré de cette notion, note immédiate, chaque erreur expliquée. Gratuit, sans inscription.

🎯 Pour t'entraîner — QCM en nombre, évaluations, explications détaillées, analyse de tes points faibles — l'abonnement Réussito (sans engagement) : via le QR code ou sur reussito.fr/abo

📖 Débloque toutes les fiches de Terminale : via le QR code ou sur reussito.fr/terminale



L'élève travaille sans les solutions sous les yeux ; vous corrigez ensemble avec ces pages. Chaque réponse est justifiée : pas besoin de connaître le cours.

Correction des exercices

Exercice 1. Division : [9, 1] et [5, 3], puis [9], [1], [5], [3] (cas de base). Fusion : [9] et [1] donnent [1, 9] ; [5] et [3] donnent [3, 5]. Fusion finale de [1, 9] et [3, 5] : comparer 1 et 3 (1 sort), puis 9 et 3 (3 sort), puis 9 et 5 (5 sort). La liste [3, 5] est alors épuisée : son seul élément restant, 9, est recopié directement. Résultat : [1, 3, 5, 9].

Exercice 2. Comparer 2 et 1 : 1 plus petit, il sort en premier. Comparer 2 et 4 : 2 plus petit, il sort ensuite. Comparer 6 et 4 : 4 plus petit, il sort ensuite ; la liste [1, 4] est alors épuisée, ses éléments restants, 6 puis 9, sont recopiés directement dans leur ordre. Résultat : [1, 2, 4, 6, 9].

Réponses du QCM — expliquées une par une

- 1 → A** Le principe diviser pour régner divise d'abord le problème en sous-problèmes, les résout (règne) récursivement, puis combine leurs résultats. *Piège : combiner en premier n'a pas de sens : il n'y a encore rien à combiner avant d'avoir résolu les sous-problèmes ; régner en premier reviendrait à résoudre le problème initial directement, sans utiliser diviser pour régner du tout.*
- 2 → A** Une liste d'un seul élément (ou vide) est déjà triée par définition : c'est le cas de base qui arrête la division récursive. *Piège : une liste de deux éléments n'est pas encore un cas de base : elle est encore divisée en deux listes d'un élément ; la fin des fusions décrit la fin de l'algorithme, pas le cas de base de la récursivité de division.*
- 3 → B** La fusion compare à chaque étape les premiers éléments restants de chaque liste et place le plus petit dans le résultat, jusqu'à épuiser les deux listes. *Piège : recopier une liste puis l'autre ne produit pas une liste triée (cf. trouve l'erreur) ; trier chaque liste séparément n'a pas de sens ici, les deux listes étant déjà triées avant la fusion.*
- 4 → C** Le nombre de niveaux de récursion ($\log n$, chaque niveau divise la taille par deux) multiplié par les n opérations de fusion par niveau donne $O(n \log n)$. *Piège : $O(n)$ correspondrait à un seul parcours de la liste, sans division récursive ; $O(n \times n)$ correspondrait à deux boucles imbriquées portant chacune sur n éléments, ce qui n'est pas la structure du tri fusion.*
- 5 → A** 16 éléments, puis 8, puis 4, puis 2, puis 1 : quatre divisions par deux sont nécessaires pour atteindre une liste d'un seul élément, soit 4 niveaux de récursion. *Piège : 8 est la taille de la liste après la première division, pas le nombre de niveaux ; 16 est la taille initiale de la liste, pas un nombre de niveaux de récursion.*
- 6 → B** Chaque appel récursif divise la liste en deux moitiés plus petites ; la taille diminue à chaque niveau jusqu'à atteindre une liste d'un seul élément, le cas de base qui arrête la récursivité. *Piège : aucun compteur externe n'est utilisé dans le tri fusion, c'est la taille décroissante des sous-listes qui arrête la récursivité ; la fusion intervient APRÈS la division, elle ne l'empêche pas.*
- 7 → A** Un sous-problème est une instance plus petite du même problème : trier une moitié de liste est bien un problème de tri, en plus petit. *Piège : la fusion est l'étape COMBINER, pas un sous-problème DIVISER ; le résultat final trié est la solution globale obtenue après avoir résolu tous les sous-problèmes, pas un sous-problème lui-même.*
- 8 → B** La fusion crée des listes temporaires proportionnelles à la taille des données traitées, ce qui donne une complexité en espace $O(n)$. *Piège : $O(1)$ correspondrait à une mémoire constante, ce qui n'est pas le cas ici ; $O(n \times n)$ correspondrait à une croissance bien plus rapide, sans rapport avec la mémoire réellement utilisée par les fusions.*

9 → C

La fusion compare toujours les premiers éléments restants de chaque liste : ici 3 (premier de la première liste) et 1 (premier de la seconde). *Piège : 8 et 5 sont les derniers éléments de chaque liste, pas les premiers à comparer ; 3 et 8 appartiennent tous les deux à la même liste, la fusion compare des éléments de listes DIFFÉRENTES.*

10 → C

Un sous-problème doit être une instance plus petite du MÊME problème pour que le principe diviser pour régner s'applique, ce qui permet de le résoudre récursivement par le même algorithme. *Piège : un problème totalement différent ne pourrait pas être résolu par le même algorithme appliqué récursivement ; le tri fusion résout ses sous-problèmes par appel récursif de lui-même, pas par une boucle.*