



SPÉ NSI · TERMINALE

✓ EXIGIBLE AU BAC

PRIORITÉ P1

Structures linéaires

listes chaînées, piles et files — opérations et complexité

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

- ✓ Décrire le fonctionnement d'une liste chaînée et de ses opérations de base.
- ✓ Distinguer le comportement d'une pile de celui d'une file, et connaître leurs opérations.
- ✓ Évaluer la complexité algorithmique des opérations sur ces trois structures linéaires.

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 !*

Liste chaînée — une structure de données linéaire où chaque élément (nœud) contient une valeur et un lien vers le nœud suivant ; l'accès à un nœud se fait en parcourant les liens un par un depuis le premier nœud.

Nœud — l'élément de base d'une liste chaînée ou d'un arbre, qui stocke une valeur et un ou plusieurs liens vers d'autres nœuds.

Pile — une structure linéaire où l'on n'ajoute et ne retire des éléments que par une seule extrémité, appelée le sommet ; le dernier élément empilé est le premier dépilé (comportement LIFO, « dernier entré, premier sorti »).

File — une structure linéaire où l'on ajoute les éléments par une extrémité (la queue) et où on les retire par l'autre extrémité (la tête) ; le premier élément enfilé est le premier défilé (comportement FIFO, « premier entré, premier sorti »).

Empiler / dépiler — les deux opérations de base d'une pile : empiler ajoute un élément au sommet, dépiler retire l'élément qui se trouve au sommet.

Enfiler / défiler — les deux opérations de base d'une file : enfiler ajoute un élément à la queue, défiler retire l'élément qui se trouve à la tête.

Complexité algorithmique — une mesure du nombre d'opérations élémentaires effectuées par un algorithme en fonction de la taille n des données traitées, dans le pire des cas.

Notation O (grand O) — la notation qui exprime une complexité algorithmique : $O(1)$ signifie un nombre d'opérations constant, indépendant de n ; $O(n)$ signifie un nombre d'opérations proportionnel à n .

Tableau indexé — une structure de données dont chaque élément est repéré par une position numérotée (un indice), ce qui permet d'y accéder directement, sans parcourir les éléments précédents.

Mémoire vive — la mémoire de l'ordinateur qui stocke temporairement les données et programmes en cours d'exécution ; sa quantité disponible est une caractéristique de la machine, pas de la complexité d'un algorithme.

2 Les règles essentielles *chaque règle avec son exemple*

- 1 **Une liste chaînée range ses éléments dans des nœuds reliés les uns aux autres ; pour atteindre un nœud quelconque, il faut parcourir les nœuds précédents un par un depuis le premier.**

exemple → Dans une liste chaînée de nœuds $3 \rightarrow 7 \rightarrow 1 \rightarrow 9$, atteindre la valeur 1 impose de passer d'abord par les nœuds 3 puis 7.

- 2 **Une pile respecte le comportement LIFO : seules deux opérations sont possibles, empiler et dépiler, et elles agissent uniquement sur le sommet de la pile.**

exemple → Si l'on empile successivement 3, 7, 1, le sommet contient 1 : un dépilement retire 1 en premier, avant 7 puis 3.

- 3 **Une file respecte le comportement FIFO : on enfile à la queue et on défile à la tête, jamais l'inverse.**

exemple → Si l'on enfile successivement 3, 7, 1 (3 en tête), un défilement retire 3 en premier, avant 7 puis 1.

- 4 **La complexité algorithmique d'une opération se détermine en comptant le nombre d'opérations élémentaires nécessaires dans le pire des cas, en fonction de la taille n de la structure, et se note en notation O.**

exemple → Retrouver le dernier nœud d'une liste chaînée de n nœuds impose de parcourir les n nœuds dans le pire des cas : la complexité est $O(n)$.

- 5 **Empiler, dépiler, enfiler et défiler s'effectuent en $O(1)$ car ces opérations agissent sur une extrémité connue directement (le sommet de la pile, ou la tête/la queue de la file), sans dépendre du nombre d'éléments déjà présents.**

exemple → Dépiler une pile de 3 éléments ou une pile de 3 000 éléments demande exactement une seule opération : dans les deux cas, la complexité est $O(1)$.

3

Méthode — Déterminer la complexité d'une opération sur une structure linéaire

Exemple suivi : évaluer la complexité de l'accès au 4e nœud d'une liste chaînée de 6 nœuds, puis celle d'un dépilement sur une pile de 6 éléments.

Étape 1 Identifier la structure utilisée (liste chaînée, pile ou file) et l'opération demandée.

Étape 2 Vérifier si l'opération agit uniquement sur une extrémité connue directement (sommet de la pile, tête ou queue de la file) : dans ce cas, elle s'effectue en $O(1)$, quel que soit le nombre d'éléments.

Étape 3 Sinon, si l'opération nécessite de parcourir les nœuds un par un pour atteindre un élément quelconque d'une liste chaînée, compter le nombre de nœuds visités dans le pire des cas : il est proportionnel à n , donc la complexité est $O(n)$. Ici, atteindre le 4e nœud impose de visiter les 4 premiers nœuds de la liste depuis le premier : la complexité est $O(n)$.

Étape 4 Pour le dépilement, la pile n'est manipulée que par son sommet, connu directement : la complexité est $O(1)$, indépendamment des 6 éléments présents.

✓ **Conclure en notant chaque complexité sous forme $O(\dots)$, en justifiant par le nombre d'opérations élémentaires effectuées.**

4

Exemple type Bac — rédigé pas à pas

Une pile contient 5 éléments empilés successivement dans cet ordre : 3, 7, 1, 9, 4 (4 empilé en dernier). On effectue un dépilement. Donner la valeur dépilée et justifier la complexité de cette opération.

Étape 1 — Identifier le sommet de la pile : le dernier élément empilé est 4, il se trouve donc au sommet.

Étape 2 — Appliquer le comportement LIFO de la pile : un dépilement retire toujours l'élément qui se trouve au sommet, ici 4.

Étape 3 — Déterminer la complexité : le dépilement agit uniquement sur le sommet, connu directement, sans parcourir les 4 autres éléments de la pile : l'opération est en $O(1)$.

Conclusion : La valeur dépilée est 4, avec une complexité $O(1)$.

5

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

✗ **Confondre** le comportement LIFO d'une pile (dernier entré, premier sorti) avec le comportement FIFO d'une file (premier entré, premier sorti).

✗ **Croire** que toute opération sur une liste chaînée est en $O(1)$: seul l'accès au premier nœud l'est ; l'accès à un nœud quelconque est en $O(n)$.

X Se contenter d'écrire « $O(1)$ » ou « $O(n)$ » sans justifier la complexité par le nombre d'opérations élémentaires réellement effectuées.

6 Tips — pour t'en souvenir

Pile = LIFO = pile d'assiettes : on pose et on retire toujours par le dessus (le sommet).

File = FIFO = file d'attente réelle : le premier arrivé est le premier servi.

Un doute sur la complexité ? Demande-toi : « cette opération dépend-elle du nombre d'éléments, ou agit-elle sur une position toujours connue ? »

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

« STRUCTURES LINÉAIRES » au centre, 3 branches colorées : LISTE CHAÎNÉE (nœuds reliés, accès $O(n)$), PILE (LIFO, empiler/dépiler au sommet, $O(1)$), FILE (FIFO, enfiler en queue/défiler en tête, $O(1)$). Un exemple chiffré 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 : « Pour retrouver le 3e nœud d'une liste chaînée de 10 nœuds, il suffit d'accéder directement à sa position, comme dans un tableau : cette opération est donc en $O(1)$. »

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

L'erreur : Une liste chaînée ne permet pas d'accéder directement à une position : pour atteindre le 3e nœud, il faut parcourir les nœuds un par un depuis le premier, ce qui est en $O(n)$ et non en $O(1)$. L'accès direct par position est une propriété des tableaux indexés, pas des listes chaînées. Dans une liste chaînée, chaque nœud ne connaît que le nœud suivant : atteindre un nœud quelconque impose de parcourir tous les nœuds précédents, d'où une complexité $O(n)$.

Exercice 1. Une file contient les éléments 2, 5, 8 enfilés dans cet ordre (2 en tête, 8 en queue). On défile un élément puis on enfila la valeur 6. Donner le contenu de la file, de la tête à la queue, et la complexité de chacune des deux opérations.

Exercice 2. Une liste chaînée contient 6 nœuds. Donner, en le justifiant, le nombre maximal de nœuds à parcourir pour accéder au dernier nœud, et la complexité correspondante.

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 comportement d'une pile est appelé :

- ☐ A LIFO (dernier entré, premier sorti)
- ☐ B FIFO (premier entré, premier sorti)
- ☐ C un accès direct par position

2. Le comportement d'une file est appelé :

- ☐ A LIFO (dernier entré, premier sorti)
- ☐ B FIFO (premier entré, premier sorti)
- ☐ C un accès direct par position

3. On empile successivement 4, 2, 9 sur une pile vide (9 en dernier). Le premier élément dépilé est :

- ☐ A 4
- ☐ B 2
- ☐ C 9

4. On enfile successivement 4, 2, 9 dans une file vide (4 en tête). Le premier élément défilé est :

- ☐ A 4
- ☐ B 2
- ☐ C 9

5. Accéder au 5e nœud d'une liste chaînée de 20 nœuds a pour complexité :

- ☐ A $O(1)$
- ☐ B $O(n)$
- ☐ C impossible à déterminer sans exécuter le programme

6. Dépiler le sommet d'une pile de 500 éléments a pour complexité :

- ☐ A $O(1)$
- ☐ B $O(n)$
- ☐ C $O(500)$ uniquement, jamais généralisable

7. Une opération qui n'est PAS possible directement sur une file est :

- ☐ A enfiler un élément à la queue
- ☐ B défiler un élément à la tête
- ☐ C retirer directement un élément situé au milieu de la file

8. Dans une liste chaînée, chaque nœud connaît :

- ☐ A sa position exacte dans la liste
- ☐ B le nœud suivant
- ☐ C l'ensemble des autres nœuds de la liste

9. La complexité algorithmique d'une opération mesure :

- ☐ A le nombre d'opérations élémentaires effectuées en fonction de la taille des données
- ☐ B le temps réel d'exécution en secondes sur un ordinateur donné
- ☐ C la quantité de mémoire vive disponible sur l'ordinateur

10. Empiler un élément sur une pile qui contient déjà 1 000 éléments a pour complexité :

- ☐ A $O(n)$
- ☐ B $O(1)$
- ☐ C $O(1000)$ uniquement pour cette pile précise



11

En 30 secondes — ta carte de dernière révision

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

- Liste chaînée : nœuds reliés en chaîne, accès à un nœud quelconque en $O(n)$ (parcours depuis le début).
- Pile (LIFO) : empiler/dépiler au sommet uniquement, en $O(1)$. File (FIFO) : enfiler à la queue, défiler à la tête, en $O(1)$.
- La complexité $O(\dots)$ compte le nombre d'opérations élémentaires en fonction de n , dans le pire des cas.

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. Défiler retire l'élément de tête (2) : la file contient 5, 8. Enfiler 6 l'ajoute en queue : la file contient alors 5, 8, 6 (de la tête à la queue). Défiler et enfiler agissent chacun sur une extrémité connue directement : les deux opérations sont en $O(1)$.

Exercice 2. Pour atteindre le dernier nœud d'une liste chaînée de 6 nœuds, il faut parcourir les nœuds un par un depuis le premier : dans le pire des cas, les 6 nœuds sont visités. Le nombre d'opérations est donc proportionnel à n (ici 6), soit une complexité $O(n)$.

Réponses du QCM — expliquées une par une

1 → A

Une pile respecte le comportement LIFO : le dernier élément empilé est le premier dépilé. *Piège : FIFO = décrit le comportement d'une file, pas d'une pile ; accès direct par position = décrit un tableau indexé, aucune de ces deux structures ne le permet.*

2 → B

Une file respecte le comportement FIFO : le premier élément enfilé est le premier défilé. *Piège : LIFO = décrit le comportement d'une pile, pas d'une file ; accès direct par position = décrit un tableau indexé.*

3 → C

Le dernier élément empilé (9) se trouve au sommet : il est donc le premier dépilé (comportement LIFO). *Piège : 4 = c'est le premier élément empilé, il serait le premier défilé dans une file, pas le premier dépilé d'une pile ; 2 = ni le premier ni le dernier empilé, il ne peut pas être en tête de dépilement.*

4 → A

Le premier élément enfilé (4) se trouve en tête de file : il est donc le premier défilé (comportement FIFO). *Piège : 2 = deuxième élément enfilé, il sera défilé en deuxième, pas en premier ; 9 = dernier élément enfilé, il se trouve en queue, ce serait le premier dépilé d'une pile, pas le premier défilé d'une file.*

5 → B

Atteindre le 5e nœud impose de visiter les 5 premiers nœuds de la liste depuis le premier : le nombre d'opérations est proportionnel à n , donc la complexité est $O(n)$. *Piège : $O(1)$ = confond l'accès à un nœud quelconque avec l'accès au premier nœud, qui lui est en $O(1)$; impossible à déterminer = la complexité se calcule justement à partir de la structure, sans exécuter le programme.*

6 → A

Dépiler agit uniquement sur le sommet, connu directement, sans parcourir les autres éléments : l'opération est en $O(1)$, quel que soit le nombre d'éléments présents. *Piège : $O(n)$ = confond avec le parcours d'une liste chaînée, alors que le sommet d'une pile est toujours connu directement ; $O(500)$ uniquement = la notation O généralise justement à toute taille n , elle ne dépend pas d'une valeur fixée.*

7 → C

Une file ne permet d'ajouter et de retirer des éléments que par ses deux extrémités (queue et tête) ; retirer un élément situé au milieu n'est pas une opération de base de la file. *Piège : enfiler à la queue et défiler à la tête sont au contraire les deux opérations de base d'une file, parfaitement possibles directement.*

8 → B

Un nœud d'une liste chaînée stocke une valeur et un lien vers le nœud suivant uniquement. *Piège : sa position exacte = propriété d'un tableau indexé, pas d'une liste chaînée ; l'ensemble des autres nœuds = surcharge inutile, un nœud ne connaît que son successeur direct.*

9 → A

La complexité algorithmique compte le nombre d'opérations élémentaires en fonction de la taille n des données, indépendamment de la machine utilisée. *Piège : le temps réel en secondes dépend de la machine et n'est pas la définition de la complexité ; la mémoire vive disponible est une caractéristique de l'ordinateur, pas de l'algorithme.*

10 → B

Empiler ajoute toujours l'élément au sommet, une position connue directement, sans dépendre du nombre d'éléments déjà présents : la complexité est $O(1)$. *Piège : $O(n)$ = confond avec une opération qui parcourrait la structure, ce que ne fait pas un empilement ; $O(1000)$ uniquement = la notation O généralise à toute taille n , ce n'est pas propre à 1000 éléments.*