

Programmation dynamique — Mémoïsation

NSI · Terminale
Contrôle n°1 · 60 min · sur 20

Exercice 1 — La condition

5 pts 14 min

Vérifier la condition de chevauchement

Réponds à chaque question.

LA RÈGLE DU COURS

Une seule propriété du problème décide si le procédé apporte quelque chose. Il faut la vérifier avant d'écrire une ligne.

1. À quelle condition la programmation dynamique apporte-t-elle quelque chose ? (1 pt)

Si les sous-problèmes se chevauchent, c'est-à-dire si les mêmes calculs sont demandés plusieurs fois.

→ C'est la seule propriété qui décide du gain.

2. Que se passe-t-il sans recouvrement ? (1 pt)

Il n'y a rien à mémoriser : chaque résultat stocké ne sert qu'une fois, et l'on paie la mémoire sans rien gagner.

→ Le programme devient plus lent et consomme davantage.

3. Quelle vérification faut-il faire au préalable ? (1 pt)

Les appels récursifs portent-ils sur des entrées déjà rencontrées ?

→ La réponse décide de tout ce qui suit.

4. Qu'ajoute-t-on exactement quand on mémoïse un problème sans recouvrement ? (1 pt)

Le coût du stockage et celui de la consultation, sans supprimer le moindre calcul.

→ La table se remplit d'entrées consultées zéro fois.

5. La mémoïsation est-elle une optimisation générale ? (1 pt)

Non.

→ Elle ne vaut que pour les problèmes dont les sous-problèmes se chevauchent.

Exercice 2 — Deux cas opposés

4 pts 11 min

Comparer deux récursions

Réponds à chaque question.

LA RÈGLE DU COURS

Deux algorithmes récursifs classiques réagissent de manière opposée au même procédé. La comparaison éclaire la condition.

1. Pourquoi le recouvrement est-il massif dans le calcul récursif de Fibonacci ? (1 pt)

Parce que le terme n demande les termes $n-1$ et $n-2$, et que chacun redemande les précédents : le terme $n-2$ est recalculé un nombre exponentiel de fois.

→ Les mêmes valeurs reviennent dans des branches différentes.

2. Que fait la mémoïsation sur Fibonacci ? (1 pt)

Elle fait chuter le coût d'exponentiel à linéaire.→ Chaque terme n est alors calculé qu'une seule fois.

3. Pourquoi le recouvrement est-il nul dans le tri fusion ? (1 pt)

Parce que chaque appel porte sur une portion différente du tableau : aucune portion n'est triée deux fois.

→ Aucun résultat mémorisé ne serait consulté.

4. Que fait la mémoïsation sur un tri fusion ? (1 pt)

Rien, sinon occuper de la mémoire.

→ La dichotomie est dans le même cas.

Exercice 3 — Deux façons de procéder

4 pts 11 min

Comparer mémorisation et tabulation

Réponds à chaque question.

LA RÈGLE DU COURS

Deux approches mettent en œuvre le même principe, l'une en gardant la récursion, l'autre en la supprimant.

1. En quoi consiste la mémorisation, et quelle est son approche ? (1 pt)

À garder la fonction récursive et lui ajouter un dictionnaire mémorisant les résultats déjà calculés ; c'est l'approche descendante.

→ Elle ne calcule que les sous-problèmes effectivement nécessaires.

2. En quoi consiste la tabulation, et quelle est son approche ? (1 pt)

À remplir un tableau des plus petits cas vers les plus grands, sans récursion ; c'est l'approche ascendante.

→ Elle évite la pile d'appels et son risque de saturation.

3. Quel inconvénient la tabulation a-t-elle ? (1 pt)

Elle calcule parfois des cases dont on n'avait pas besoin.

→ L'approche descendante, elle, ne calcule que le nécessaire.

4. Quelle difficulté propre la tabulation demande-t-elle de résoudre ? (1 pt)

Trouver le bon ordre de remplissage, chaque case ne devant dépendre que de cases déjà remplies.

→ C'est parfois le vrai travail du problème.

Exercice 4 — Deux problèmes de référence

4 pts 13 min

Analyser les problèmes de référence

Réponds à chaque question.

LA RÈGLE DU COURS

Deux problèmes classiques relèvent du même schéma, et une propriété commune les y autorise.

1. En quoi consiste le problème du sac à dos ? (1 pt)

Choisir parmi des objets ayant chacun un poids et une valeur, de façon à maximiser la valeur totale sans dépasser une capacité.

→ Les sous-problèmes portent sur les i premiers objets et une capacité donnée.

2. Quel autre problème relève du même schéma, et à quoi sert-il ? (1 pt)

La plus longue sous-séquence commune à deux chaînes ; elle sert à comparer des textes ou des séquences biologiques.

→ Les deux problèmes ont un chevauchement massif.

3. Quelle propriété ces deux problèmes partagent-ils ? (1 pt)

Une solution optimale se compose de solutions optimales de sous-problèmes.

→ C'est elle qui, avec le chevauchement, autorise le procédé.

4. Cette propriété est-elle automatique ? (1 pt)

Non : elle se vérifie.

→ Un problème où un choix localement optimal interdit le meilleur choix global ne s'y prête pas.

Repérer une erreur et la corriger

Chaque affirmation est fausse. Écris ce qu'il fallait dire.

LA RÈGLE DU COURS

Trois fautes reviennent. Une table inutile, une approche crue équivalente, et une propriété tenue pour acquise.

1. « J'ajoute une mémorisation à toutes mes fonctions récursives pour les accélérer. » (1 pt)

Sans chevauchement des sous-problèmes, la table se remplit d'entrées consultées zéro fois.

→ Le programme devient plus lent et consomme davantage.

2. « La tabulation et la mémorisation calculent exactement les mêmes cases. » (1 pt)

La tabulation calcule parfois des cases dont on n'avait pas besoin.

→ La mémorisation ne calcule que les sous-problèmes effectivement nécessaires.

3. « Puisque le problème se découpe en sous-problèmes, la programmation dynamique s'applique. » (1 pt)

Encore faut-il qu'une solution optimale se compose de solutions optimales de sous-problèmes.

→ Cette propriété se vérifie, elle n'est pas automatique.

Barème — ce que le correcteur regarde

Les lignes chiffrées couvrent les 5 exercices et totalisent 20 points.

EXERCICE	ATTENDU	POINTS	POINT PERDU SI
Ex.1	La condition de chevauchement	5 points — 1 par question	Une mémorisation ajoutée sans vérifier le recouvrement.
Ex.2	Fibonacci et tri fusion	4 points — 1 par question	Les deux cas traités de la même façon.
Ex.3	Mémorisation et tabulation	4 points — 1 par question	Les deux approches présentées comme identiques.
Ex.4	Deux problèmes de référence	4 points — 1 par question	La sous-structure optimale tenue pour acquise.
Ex.5	L'erreur repérée, et la réponse juste donnée	3 points — 1 par ligne	Une correction qui garde la table inutile.

ET MAINTENANT ?**16 à 20**

Le chapitre est acquis. Entraîne-toi maintenant à transformer une récursion en cinq temps : la version naïve écrite, le chevauchement vérifié, la table consultée avant et écrite après, le gain estimé, et la tabulation si la profondeur est un risque.

10 à 15

Refais l'exercice 2 en te demandant chaque fois si les mêmes paramètres reviennent dans plusieurs branches.

moins de 10

Reprends la fiche de cours et retiens la vérification préalable : les appels récursifs portent-ils sur des entrées déjà rencontrées ?