

EXERCICES

Tle

Programmation dynamique — Mémoïsation

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.

● ○ ○

- a. Quelle condition rend la programmation dynamique utile ?
- b. Différencier mémoïsation et tabulation.
- c. Quel risque la tabulation évite-t-elle ?
- d. Énoncer le problème du sac à dos.
- e. Quelle seconde propriété le problème doit-il vérifier ?

.....

.....

.....

.....

.....

.....

2 Un développeur ajoute une mémoïsation à une fonction de tri fusion. Analyser.

● ● ●

- a. Les sous-problèmes d'un tri fusion se chevauchent-ils ?
- b. Que contient alors la table de mémoïsation ?
- c. Le programme devient-il plus rapide ?
- d. Quel est le coût ajouté ?
- e. Quelle vérification aurait évité ce travail ?

.....

.....

.....

.....

.....

.....

3 Le calcul récursif naïf de Fibonacci recalcule massivement les mêmes valeurs.

● ● ○

- a. Pourquoi le terme $n-2$ est-il recalculé de nombreuses fois ?
- b. Combien de sous-problèmes distincts existe-t-il pour calculer le terme n ?
- c. Quelle complexité obtient-on avec mémoïsation ?
- d. Pourquoi ce gain est-il si spectaculaire ?

.....

.....

.....

.....

.....

.....

4 Mémoïsation ou tabulation. Raisonner sur le choix.



- a. Quel avantage la mémoïsation a-t-elle sur le nombre de calculs ?
- b. Quel avantage la tabulation a-t-elle sur la mémoire d'exécution ?
- c. Quelle difficulté propre la tabulation introduit-elle ?
- d. Pourquoi le sac à dos se prête-t-il au paradigme ?
- e. Quel type de problème ne s'y prête pas ?

Corrigé détaillé sur progressschool.fr — chapitre « Programmation dynamique — Mémoïsation ».