

1 Présentation

La **programmation dynamique** est un procédé algorithmique de **recherche de solution optimale** à un problème.

Exemples :

- Comment aller de A à B le plus rapidement possible ? (ex : algorithme de Bellman-Ford pour le routage RIP).
- Comment réaliser une tâche en consommant le moins de ressources possible ?
- Comment rendre une somme d'argent avec un minimum de pièces ?

L'idée générale est de décomposer le problème complet en sous-problèmes similaires plus simples, et on essaie d'abord de résoudre ces sous-problèmes pour reconstruire la solution complète, en optimisant chacun des sous-problèmes.

Conceptualisé par Bellman dans les années 1950, ici le terme « programmation » s'entend plus au sens d'organisation, d'ordonnancement que de programmation au sens langage de programmation.

Exemples :

- Pour aller de A à B le plus rapidement possible : on cherchera à atteindre des étapes intermédiaires le plus rapidement possible jusqu'à arriver à la destination finale B.
- Pour réaliser une tâche en consommant le moins de ressources possible : on découpe la tâche en plusieurs tâches intermédiaires et on ne cherche comment réaliser une nouvelle tâche que lorsqu'on a résolu le problème pour la tâche précédente.
- Pour rendre une somme d'argent avec un minimum de pièces : on cherche comment rendre 1€ avec un minimum de pièces, puis 2€, etc, jusqu'à atteindre la somme totale.

Dans la programmation dynamique, on résout donc un problème complexe par une **approche ascendante** (bottom-up) en commençant par des sous-problèmes simples pour aller atteindre la globalité du problème complexe. **La démarche mise en œuvre est itérative.**

Les solutions des sous-problèmes sont **souvent consommatrices de mémoire** car il faut conserver chacune de ces solutions jusqu'à obtenir la solution globale du problème.

2 Mémoïsation

La **mémoïsation**, qui est parfois associée à la programmation dynamique, a une **approche descendante** (top-down), du haut vers le bas. (selon moi, ce n'est pas de la programmation dynamique.)

La démarche est **récursive**.

Ici, il s'agit d'enregistrer des solutions intermédiaires calculées au cours des appels récursifs pour ne pas reproduire de multiples fois ces calculs s'ils doivent apparaître dans des appels récursifs ultérieurs.

3 Exemples

Nous verrons en TP quelques exemples d'applications.

Alignement de séquences (sujet de GO ?) : le site interstices.info en donne une bonne présentation : « Alignement optimal et comparaison de séquences génomiques et protéiques » :

<https://urlz.fr/gmsV>