

---

# Feuille d'exercices n°0 - Révisions de programmation dynamique

---

## Notions abordées

- facteur, sous-mot, préfixe, suffixe
- programmation dynamique
- pseudo-code d'algorithmes

## Exercice 1 : Plus long sous-mot/facteur commun

- Q. 1** Rappeler les définitions de **facteur** d'un mot et **sous-mot** d'un mot.  
Donner un exemple illustrant la différence entre les deux.
- Q. 2** Définir le problème du plus long sous-mot commun (PLSMC).
- Q. 3** Définir le problème du plus long facteur commun (PLFC).
- Q. 4** Expliquer comment résoudre le problème PLSMC par programmation dynamique.
- Q. 5** Expliquer comment résoudre le problème PLFC par programmation dynamique.

## 🔑 Éléments pour établir un algorithme de programmation dynamique

- identifier une famille de "sous-problèmes"<sup>a</sup> qu'il est intéressant de résoudre ;
- identifier quels paramètres permettent de caractériser ces sous-problèmes relativement à l'instance de départ ;
- définir une quantité paramétrée qui représente la valeur optimale du sous-problème défini par ces paramètres ;
- justifier que cette famille de quantités permet de résoudre le problème initial ;
- dire comment calculer les cas de base, *i.e.* les quantités pour les paramètres minimaux ;
- dire comment calculer l'une de ces quantités à partir des valeurs pour des paramètres plus petits ;
- si on opte pour de l'impératif, fournir le pseudo-code qui explique quelle taille de tableau alouer pour stocker ces quantités, dans quel ordre on va remplir le tableau, comment on va ensuite récupérer la valeur optimale ;
- dans le cas où seule la valeur optimale nous intéresse, il est souvent possible de modifier l'algorithme précédent en vue de réduire sa consommation mémoire ;
- dans le cas où une solution optimale doit aussi être fournie, on complète l'algorithme précédent en ajoutant le plus souvent l'enregistrement d'information pertinente lors du remplissage du tableau, et en ajoutant une phase de reconstruction de la solution qui remonte dans le tableau à partir d'une case de valeur optimale.

---

<sup>a</sup>. On appelle ici, comme c'est l'usage, famille de sous-problèmes une famille d'instances, instances du problème initial ou parfois d'une variante de celui-ci.