

Anagramme : Programmation dynamique

Vérifiez les conventions à suivre pour l'écriture du code (cf Moodle).

L'anagramme d'un mot est un autre mot qui utilise les mêmes lettres mais dans un sens différent : on étend cette définition aux mots qui n'ont pas de sens particuliers.

Par exemple, le mot 'abc' possède les anagrammes 'acb', 'bac', 'cab', 'cba' : en fait, toutes les combinaisons et positions de lettres possibles du mot de départ.

En informatique, la recherche d'anagramme est une manière d'énumérer les possibilités de codage ; ce qui est extrêmement utile en recherche de mots de passe mais également dans tous les algorithmes de recherche de solution sans heuristique (cf TD Metro).

Vous allez créer deux fonctions qui permettent de rechercher tous les anagrammes d'un mot particulier :

```
public static Set<String> trouveAnagrammesDe(String mot_) { ... }
```

Vous pourrez créer une autre fonction utilitaire, récursive, qui est une fonction *pure* – c'est à dire qui n'utilise pas de variables extérieure, pas d'entrées/sorties et ne modifie pas ses paramètres.

Vous pourrez programmer en Java, ou en Scala (si vous vous en sentez capable).

Combien trouvez-vous d'anagrammes pour 'abcdefg' ? et pour 'elephant' ?