

# Programmation dynamique et alignement de séquence

- Alignement de séquences : la plus longue sous-chaîne commune
- Avant cela : à quelle distance se trouve-t-on du mot correct ?

↓ ↓  
AGORRYTNES  
ALGORITHMES

Distance 5

AGORRYTNES → ALGORRYTNES → ALGORYTNES  
ALGORITHMES ← ALGORITHNES ← ALGORITNES

# La distance d'édition

- La distance d'édition (ou de Levenshtein, ou d'Ulam) entre deux mots A et B est la longueur de la plus courte suite de transformations pour passer de A à B, avec les transformations suivantes :
  - insertion d'une nouvelle lettre
  - suppression d'une lettre
  - remplacement d'une lettre par une autre
- Définition alternative : Nombre minimal de désaccords dans un alignement de A et B

A	_	G	O	R	R	Y	T	_	N	E	S
A	L	G	O	_	R	I	T	H	M	E	S

# Distance d'édition

- Entrée
  - Deux mots A et B sur un alphabet (mot : chaîne de caractère ou tableau de caractères ou ...)
- Sortie 1
  - La distance d'édition entre A et B
- Sortie 2
  - Une suite de transformations minimale de A à B

# Distance d'édition

Utilité :

- Orthographe :
  - Correcteur orthographique
  - Reconnaissance optique de caractères
- Linguistique (proximité de langues)
- Bioinformatique :
  - similarité de séquences ADN
  - similarité d'arbres phylogénétiques
- . . .



## Algorithme – EDITION

Entrées : Deux mots  $u$  et  $v$

Sortie : la distance d'édition entre  $u$  et  $v$

1  $m \leftarrow$  taille de  $u$ ;  $n \leftarrow$  taille de  $v$

2  $E \leftarrow$  tableau bidimensionnel de dimensions  $(m + 1) \times (n + 1)$

### # Cas de base

3 Pour  $i = 0$  à  $m$  :  $E_{i,0} = i$

4 Pour  $j = 0$  à  $n$  :  $E_{0,j} = j$

### # Formule récursive

5 Pour  $i = 1$  à  $m$  :

6   Pour  $j = 1$  à  $n$  :

7      $E_{i,j} \leftarrow \min \left( E_{i,j-1} + 1, E_{i-1,j} + 1, E_{i-1,j-1} + \begin{matrix} 1 & \text{si } u[i-1] \neq v[j-1] \\ 0 & \text{sinon} \end{matrix} \right)$

8 Renvoyer  $E_{m,n}$