

La méthode des 4 Russes^a :

- En 1970, quatre mathématiciens russes, Arlazarov, Dinic, Kronrod et Faradzev, publient un algorithme permettant de calculer les entrées d'une matrice de dimension $n \times n$, reliée au calcul de la fermeture transitive d'un graphe orienté sur n sommets, en temps $\mathcal{O}(n^2/\log n)$

^a Arlazarov, V.L., Dinic, E.A., Kronrod, M.A. et Faradzev, I.A., *On economic construction of the transitive closure of a directed graph*, Dokl. Akad. Nauk. SSSR, 194, pp. 487–488 1970 (en Russe). Traduction anglaise dans *Soviet Math. Dokl.*, 11, pp. 1209–1210, 1975.

La méthode des 4 Russes^a :

- En 1970, quatre mathématiciens russes, Arlazarov, Dinic, Kronrod et Faradzev, publient un algorithme permettant de calculer les entrées d'une matrice de dimension $n \times n$, reliée au calcul de la fermeture transitive d'un graphe orienté sur n sommets, en temps $\mathcal{O}(n^2/\log n)$
- Un nombre l est choisi assez petit et divisant parfaitement n . Ensuite, on subdivise la grande matrice $n \times n$ en sous-matrices de dimension $l \times l$ et on se sert des résultats précalculés pour trouver le résultat final.

^a Arlazarov, V.L., Dinic, E.A., Kronrod, M.A. et Faradzev, I.A., *On economic construction of the transitive closure of a directed graph*, Dokl. Akad. Nauk. SSSR, 194, pp. 487–488 1970 (en Russe). Traduction anglaise dans *Soviet Math. Dokl.*, 11, pp. 1209–1210, 1975.

La méthode des 4 Russes^a :

- En 1970, quatre mathématiciens russes, Arlazarov, Dinic, Kronrod et Faradzev, publient un algorithme permettant de calculer les entrées d'une matrice de dimension $n \times n$, reliée au calcul de la fermeture transitive d'un graphe orienté sur n sommets, en temps $\mathcal{O}(n^2/\log n)$
- Un nombre l est choisi assez petit et divisant parfaitement n . Ensuite, on subdivise la grande matrice $n \times n$ en sous-matrices de dimension $l \times l$ et on se sert des résultats précalculés pour trouver le résultat final.
- Cette méthode est communément appelée la *méthode des 4 Russes*.

^a Arlazarov, V.L., Dinic, E.A., Kronrod, M.A. et Faradzev, I.A., *On economic construction of the transitive closure of a directed graph*, Dokl. Akad. Nauk. SSSR, 194, pp. 487–488 1970 (en Russe). Traduction anglaise dans *Soviet Math. Dokl.*, 11, pp. 1209–1210, 1975.

Algo. de Masek-Paterson (Alig. global)^a :

- En 1980, Masek et Paterson ont l'idée de se servir de la méthode des 4 Russes pour résoudre le problème de l'alignement global de deux mots en temps $\mathcal{O}(n^2/\log n)$.

^aMasek, W.J. et Paterson, M.S., *A Faster Algorithm Computing String Edit Distance*, *Journal of Computer and System Sciences*, 20, pp. 18–31, 1980.

Algo. de Masek-Paterson (Alig. global)^a :

- En 1980, Masek et Paterson ont l'idée de se servir de la méthode des 4 Russes pour résoudre le problème de l'alignement global de deux mots en temps $\mathcal{O}(n^2/\log n)$.
- Étant donné deux mots de longueur n , ils commencent par choisir un petit nombre k , tel que k divise n . Ils précalculent ensuite toutes les sous-matrices possibles de dimension $(k + 1) \times (k + 1)$.

^aMasek, W.J. et Paterson, M.S., *A Faster Algorithm Computing String Edit Distance*, *Journal of Computer and System Sciences*, 20, pp. 18–31, 1980.

Algo. de Masek-Paterson (Alig. global)^a :

- En 1980, Masek et Paterson ont l'idée de se servir de la méthode des 4 Russes pour résoudre le problème de l'alignement global de deux mots en temps $\mathcal{O}(n^2/\log n)$.
- Étant donné deux mots de longueur n , ils commencent par choisir un petit nombre k , tel que k divise n . Ils précalculent ensuite toutes les sous-matrices possibles de dimension $(k + 1) \times (k + 1)$.
- Le calcul d'une de ces sous-matrices est entièrement déterminé par sa première ligne L , sa première colonne C et deux mots, m_1 et m_2 , de longueur k sur l'alphabet Σ considéré.

^aMasek, W.J. et Paterson, M.S., *A Faster Algorithm Computing String Edit Distance*, *Journal of Computer and System Sciences*, 20, pp. 18–31, 1980.

Algo. de Masek-Paterson (suite):

- Étant donné L , C , m_1 et m_2 fixés, on garde en mémoire la dernière ligne et la dernière colonne de la sous-matrice des distances correspondante.

Algo. de Masek-Paterson (suite):

- Étant donné L , C , m_1 et m_2 fixés, on garde en mémoire la dernière ligne et la dernière colonne de la sous-matrice des distances correspondante.
- L'algorithme se termine par le calcul de la matrice des distances D , sous-matrices par sous-matrices, au lieu de cellule par cellule.

Algo. de Masek-Paterson (suite):

- Étant donné L , C , m_1 et m_2 fixés, on garde en mémoire la dernière ligne et la dernière colonne de la sous-matrice des distances correspondante.
- L'algorithme se termine par le calcul de la matrice des distances D , sous-matrices par sous-matrices, au lieu de cellule par cellule.
- Mais combien de sous-matrices doit-on précalculer avec cette méthode?

Combien de sous-matrices?

- Les lignes et les colonnes d'une table de distances D , entre deux mots de longueur n , comprennent des valeurs entre 0 et n . De plus, deux entrées successives, sur une même ligne ou une même colonne, diffèrent toujours de 0, 1 ou -1. On a donc environ $n \cdot 3^k$ premières lignes ou colonnes possibles dans des matrices de dimension $(k + 1) \times (k + 1)$.

Combien de sous-matrices?

- Les lignes et les colonnes d'une table de distances D , entre deux mots de longueur n , comprennent des valeurs entre 0 et n . De plus, deux entrées successives, sur une même ligne ou une même colonne, diffèrent toujours de 0, 1 ou -1. On a donc environ $n \cdot 3^k$ premières lignes ou colonnes possibles dans des matrices de dimension $(k + 1) \times (k + 1)$.
- De plus, le nombre de mots de longueur k sur un alphabet Σ est $|\Sigma|^k$.

Combien de sous-matrices?

- Les lignes et les colonnes d'une table de distances D , entre deux mots de longueur n , comprennent des valeurs entre 0 et n . De plus, deux entrées successives, sur une même ligne ou une même colonne, diffèrent toujours de 0, 1 ou -1. On a donc environ $n \cdot 3^k$ premières lignes ou colonnes possibles dans des matrices de dimension $(k + 1) \times (k + 1)$.
- De plus, le nombre de mots de longueur k sur un alphabet Σ est $|\Sigma|^k$.
- On doit donc précalculer $n^2 \cdot 3^{2k} \cdot |\Sigma|^{2k}$ matrices de dimension $(k + 1) \times (k + 1)$.

Combien de sous-matrices?

- Les lignes et les colonnes d'une table de distances D , entre deux mots de longueur n , comprennent des valeurs entre 0 et n . De plus, deux entrées successives, sur une même ligne ou une même colonne, diffèrent toujours de 0, 1 ou -1. On a donc environ $n \cdot 3^k$ premières lignes ou colonnes possibles dans des matrices de dimension $(k + 1) \times (k + 1)$.
- De plus, le nombre de mots de longueur k sur un alphabet Σ est $|\Sigma|^k$.
- On doit donc précalculer $n^2 \cdot 3^{2k} \cdot |\Sigma|^{2k}$ matrices de dimension $(k + 1) \times (k + 1)$.
- Chacune de ces matrices est calculée en temps $\mathcal{O}(k^2)$.

Exemple:

- Supposons que l'on veuille calculer la distance d'édition entre les deux séquences *GTCAGG* et *CATAGT*.

Exemple:

- Supposons que l'on veuille calculer la distance d'édition entre les deux séquences *GTCAGG* et *CATAGT*.
- Ici la longueur des mots est 6 et on peut prendre, par exemple, $k = 3$.

Exemple:

- Supposons que l'on veuille calculer la distance d'édition entre les deux séquences *GTCAGG* et *CATAGT*.
- Ici la longueur des mots est 6 et on peut prendre, par exemple, $k = 3$.
- La première étape, que je ne ferai pas ici, consiste à précalculer toutes les sous-matrices de dimension 4×4 possibles sur l'alphabet $\{A, C, G, T\}$ (Il y a $6^2 \cdot 3^6 \cdot 4^6 = 107\,495\,424$)

Exemple:

- Supposons que l'on veuille calculer la distance d'édition entre les deux séquences *GTCAGG* et *CATAGT*.
- Ici la longueur des mots est 6 et on peut prendre, par exemple, $k = 3$.
- La première étape, que je ne ferai pas ici, consiste à précalculer toutes les sous-matrices de dimension 4×4 possibles sur l'alphabet $\{A, C, G, T\}$ (Il y a $6^2 \cdot 3^6 \cdot 4^6 = 107\,495\,424$)
- Garder en mémoire la dernière ligne et la dernière colonne de chacune de ces sous-matrices.

Exemple(suite): Initialisons la table des distances pour ce problème et subdivisons-la en matrices de dimension 4×4 :

	λ	G	T	C	A	G	G
λ	0	1	2	3	4	5	6
C	1						
A	2						
T	3						
A	4						
G	5						
T	6						

Exemple(suite): On commence par aller chercher la dernière ligne et la dernière colonne de la sous-matrice en haut à gauche, c'est-à-dire la sous-matrice ayant comme première ligne 0123, comme première colonne 0123 et dont les sous-mots à aligner sont *GTC* et *CAT*. On obtient ce qui suit:

	λ	<i>G</i>	<i>T</i>	<i>C</i>	<i>A</i>	<i>G</i>	<i>G</i>
λ	0	1	2	3	4	5	6
<i>C</i>	1			2			
<i>A</i>	2			3			
<i>T</i>	3	3	2	3			
<i>A</i>	4						
<i>G</i>	5						
<i>T</i>	6						

Exemple(suite): On a maintenant l'information nécessaire au calcul des dernières lignes et colonnes de deux nouvelles sous-matrices. La première correspond aux vecteurs 3456, 3233 et aux sous-mots *AGG* et *CAT* et la deuxième aux vecteurs 3323 et 3456 et aux sous-mots *GTC* et *AGT*:

	λ	<i>G</i>	<i>T</i>	<i>C</i>	<i>A</i>	<i>G</i>	<i>G</i>
λ	0	1	2	3	4	5	6
<i>C</i>	1			2			5
<i>A</i>	2			3			4
<i>T</i>	3	3	2	3	3	3	4
<i>A</i>	4			3			
<i>G</i>	5			4			
<i>T</i>	6	5	4	5			

Exemple(suite): Finalement, on va chercher l'information correspondant à la dernière sous-matrice, en bas à droite. On obtient alors que la distance d'édition entre *GTCAGG* et *CATAGT* est 4::

	λ	<i>G</i>	<i>T</i>	<i>C</i>	<i>A</i>	<i>G</i>	<i>G</i>
λ	0	1	2	3	4	5	6
<i>C</i>	1			2			5
<i>A</i>	2			3			4
<i>T</i>	3	3	2	3	3	3	4
<i>A</i>	4			3			4
<i>G</i>	5			4			4
<i>T</i>	6	5	4	5	5	4	4

Idée pour réduire le nombre de sous-matrices:

- L'idée, pour diminuer le nombre de sous-matrices de dimension $(k + 1) \times (k + 1)$ à calculer, est de travailler avec des vecteurs de différences horizontales ($D(i, j) - D(i, j - 1)$) et de différences verticales ($D(i, j) - D(i - 1, j)$) au lieu de travailler avec des lignes et des colonnes pouvant apparaître dans la table D .

Exemple:

La sous-matrice suivante de la table des distances de l'exemple précédent était représentée par sa première ligne 3456, sa première colonne 3233 ainsi que les sous-mots *AGG* et *CAT*:

	λ	<i>A</i>	<i>G</i>	<i>G</i>
λ	3	4	5	6
<i>C</i>	2			
<i>A</i>	3			
<i>T</i>	3			

Nous la représenterons maintenant par les mêmes sous-mots *AGG* et *CAT* et par le vecteur de différences horizontales 111, associé à la ligne 3456, et le vecteur de différences verticales -110, associé à la colonne 3233.

Idée (suite):

- Étant donné un vecteur de différences horizontales, correspondant à la première ligne d'une sous-matrice de distances de dimension $(k + 1) \times (k + 1)$, un vecteur de différences verticales, correspondant à la première colonne de la même sous-matrice, et deux mots m_1 et m_2 sur un alphabet Σ , Masek et Paterson donne un algorithme pour le calcul, en temps $\mathcal{O}(k^2)$, du vecteur de différences horizontales, correspondant à la dernière ligne de la sous-matrice, et du vecteur de différences verticales, correspondant à la dernière colonne de la sous-matrice
- Comme il y a 3^k vecteur de différences possibles, le nombre de sous-matrices à précalculer est réduit à $3^{2k} \cdot |\Sigma|^{2k}$.