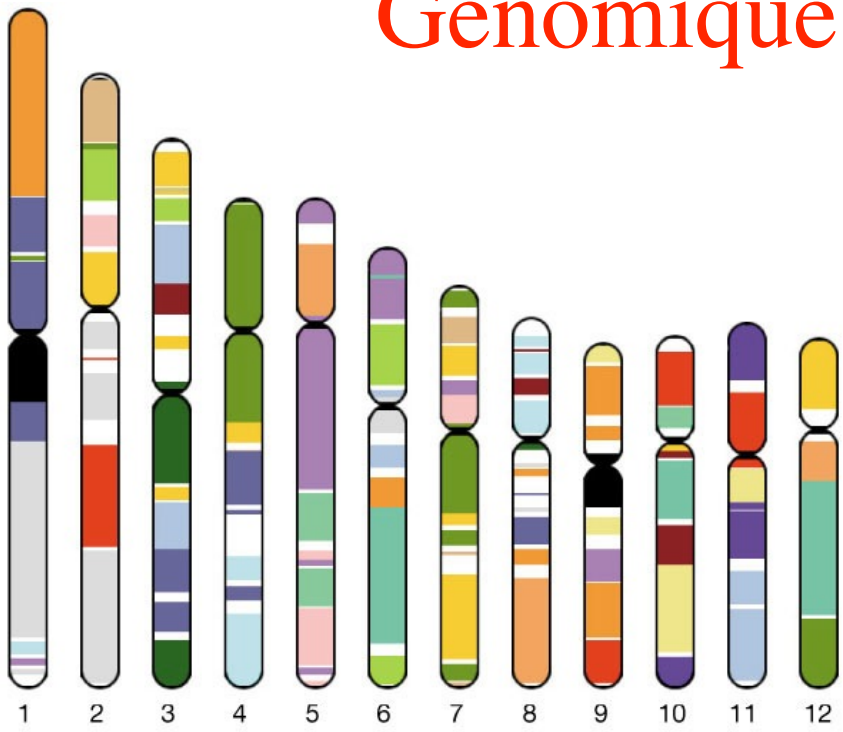
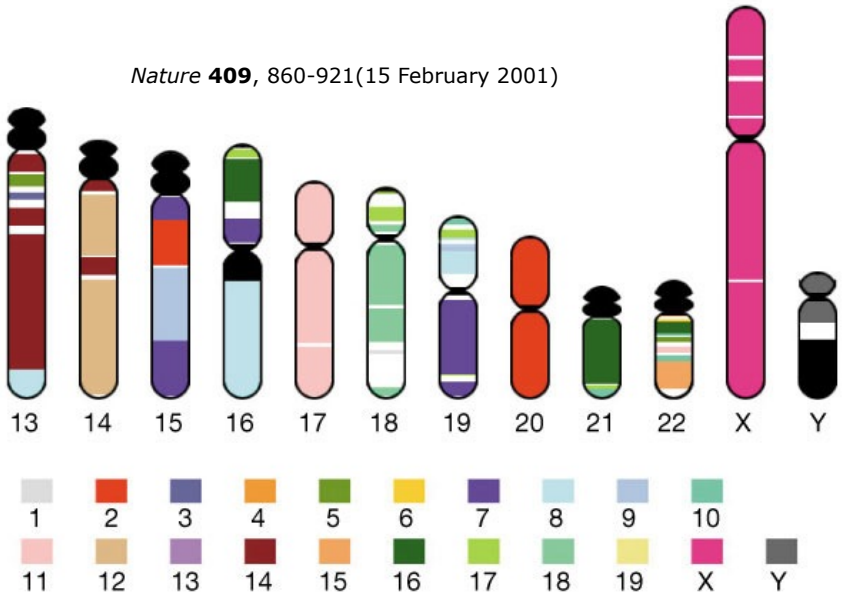
Génomique Comparative

Définition (wiki): La génomique comparative est l'étude comparative de la structure et fonction des génomes de différentes espèces. Elle permet d'identifier et de comprendre les effets de la sélection sur l'organisation et l'évolution des génomes.

Génomique Comparative

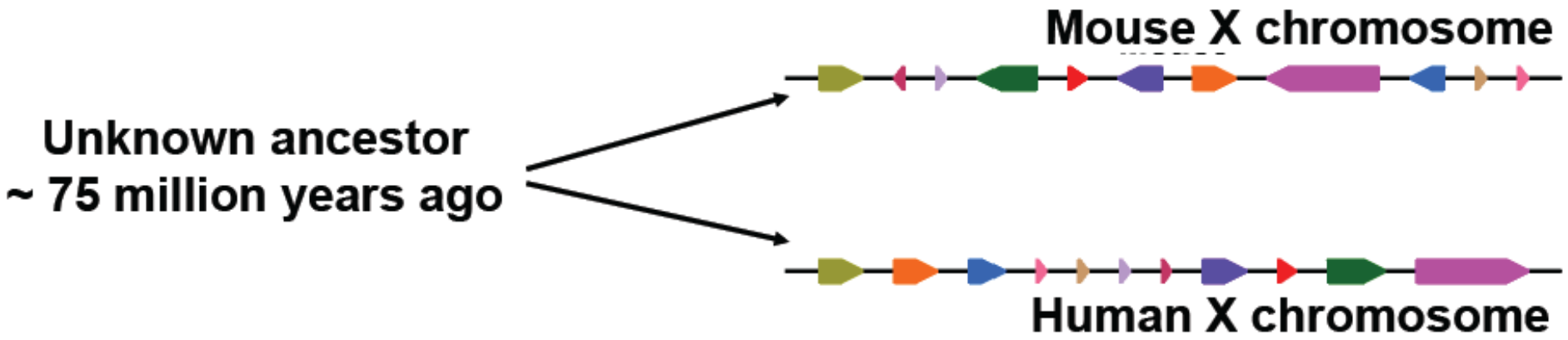


Nature **409**, 860-921(15 February 2001)



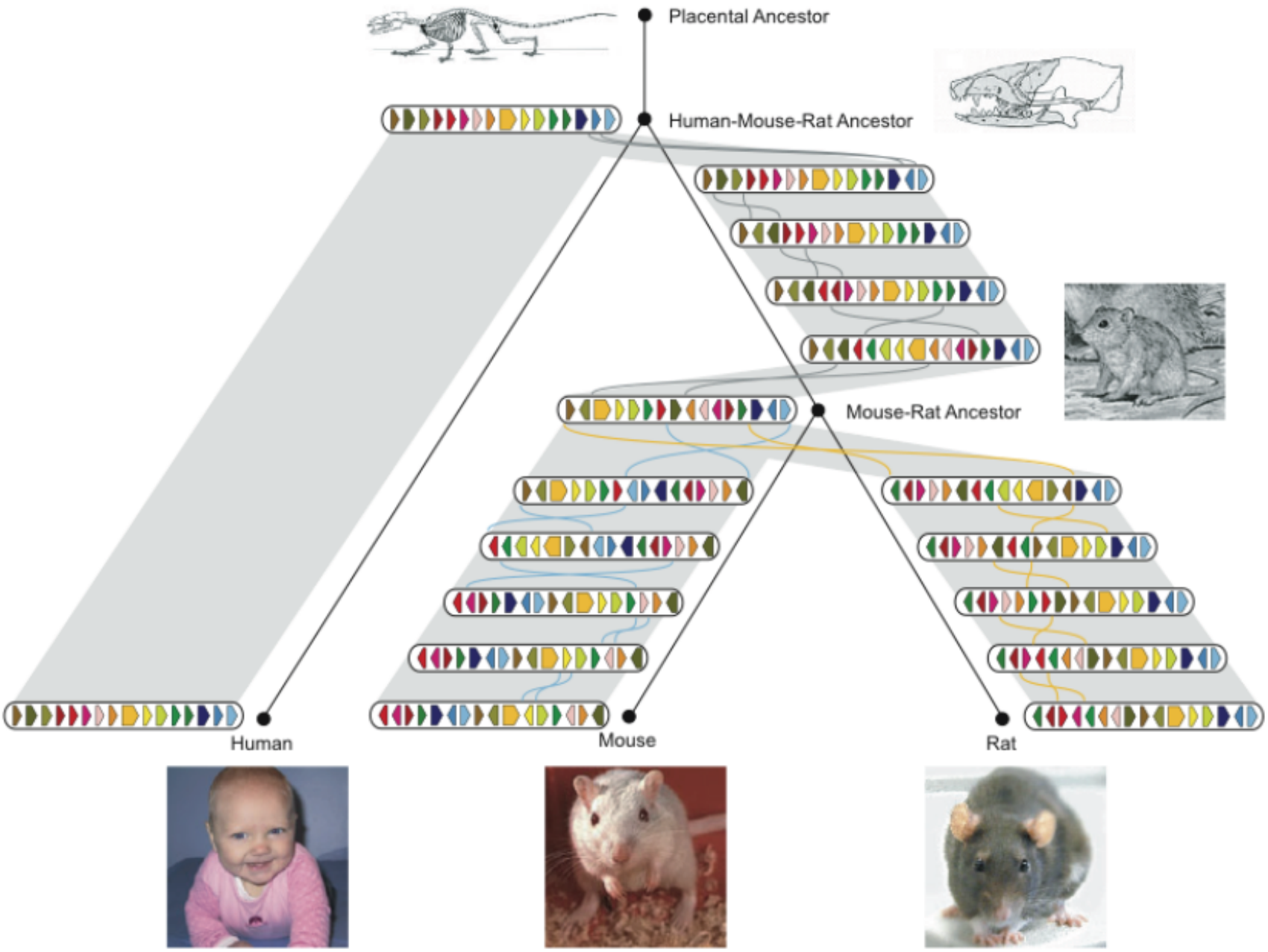
Gènes conservés entre l'homme et la souris

Génomique Comparative



www.bioalgorithms.info

Question: Peut-on expliquer le scénario d'évolution qui a permis ce changement d'ordre et d'orientation des gènes entre ces deux chromosomes?



Rat Consortium, *Nature*, 2004

Différents types de mutations génomiques

1) Réarrangements intra-chromosomales

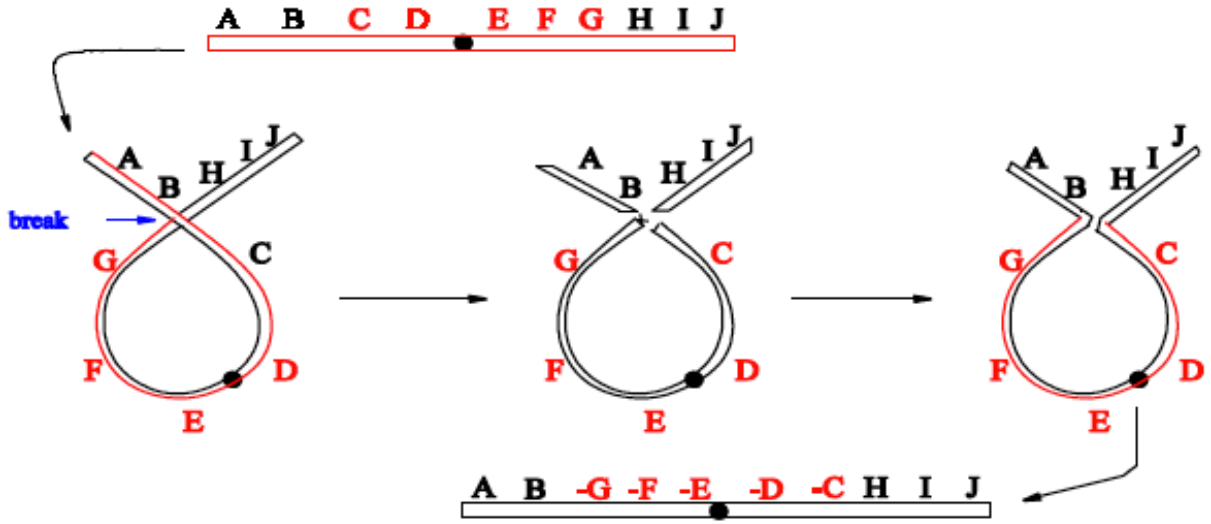
a) - Inversion:

a b | c d e | f g h i j

↓

a b -e -d -c f g h i j

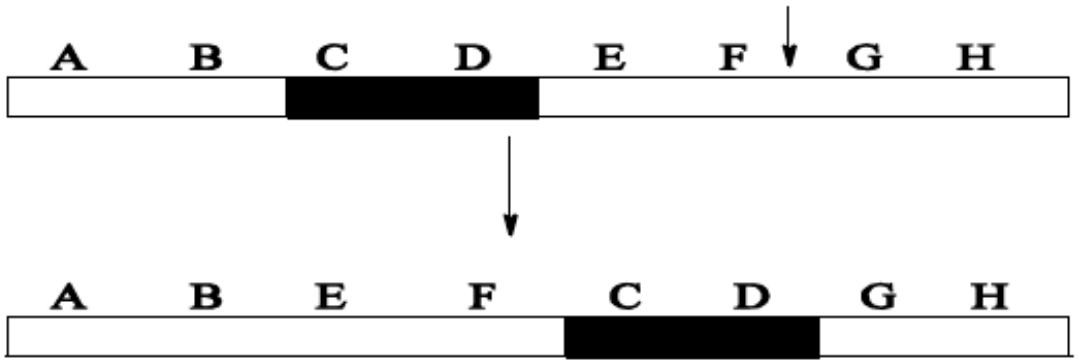
Origine possible:
Erreur de réplication



Différents types de mutations génomiques

1) Réarrangements intra-chromosomales

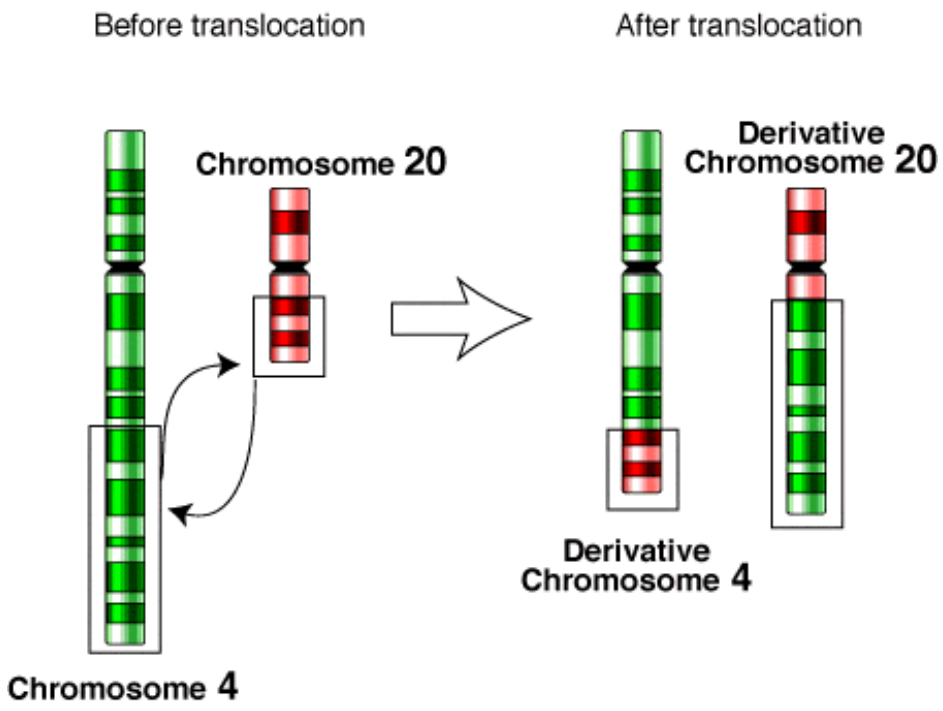
b) ▪ Transposition: Segment supprimé et réinséré à un autre endroit dans le génome



Différents types de mutations génomiques

2) Réarrangements extra-chromosomales

a) Translocation

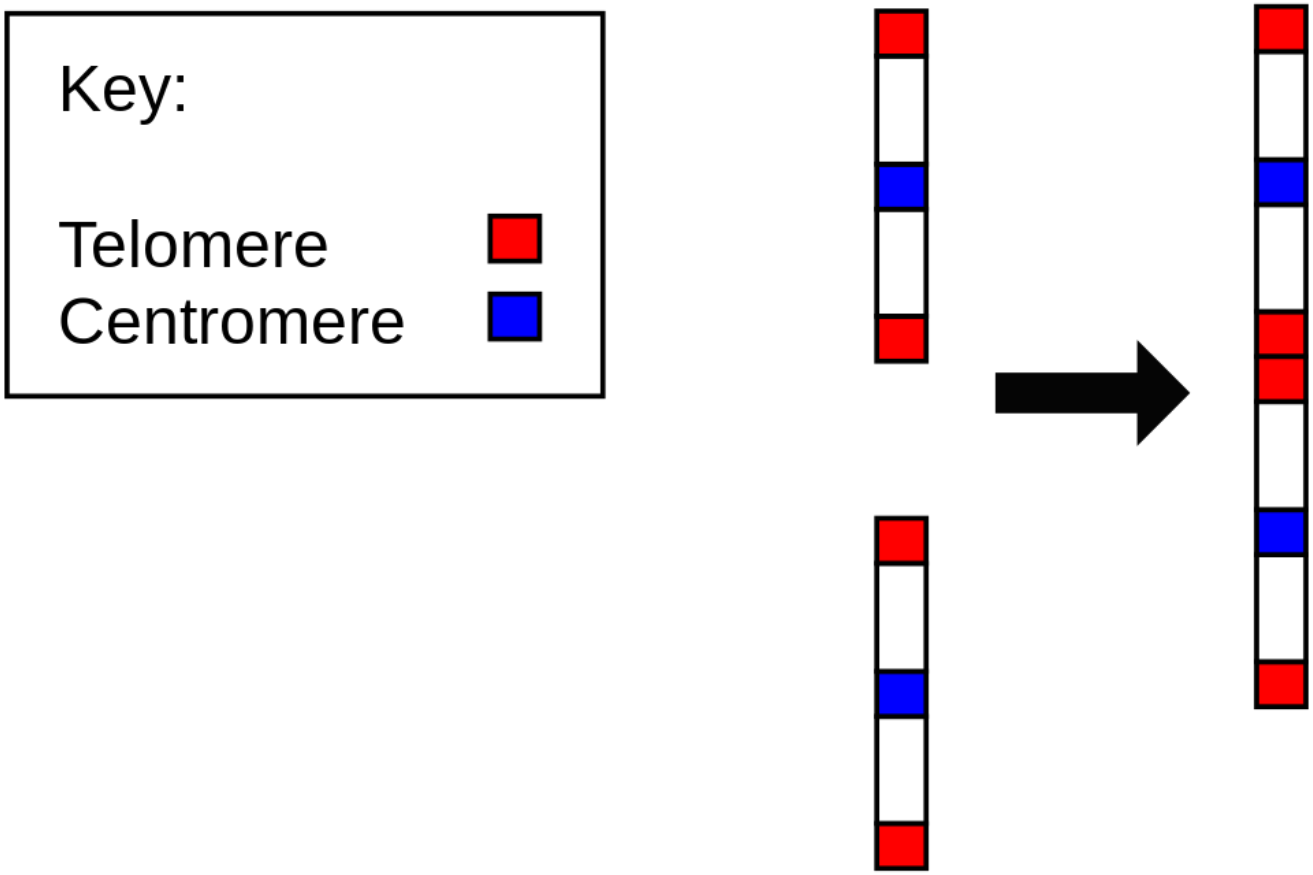


Wiki:Courtesy: National Human Genome Research Institute

Différents types de mutations génomiques

2) Réarrangements extra-chromosomales

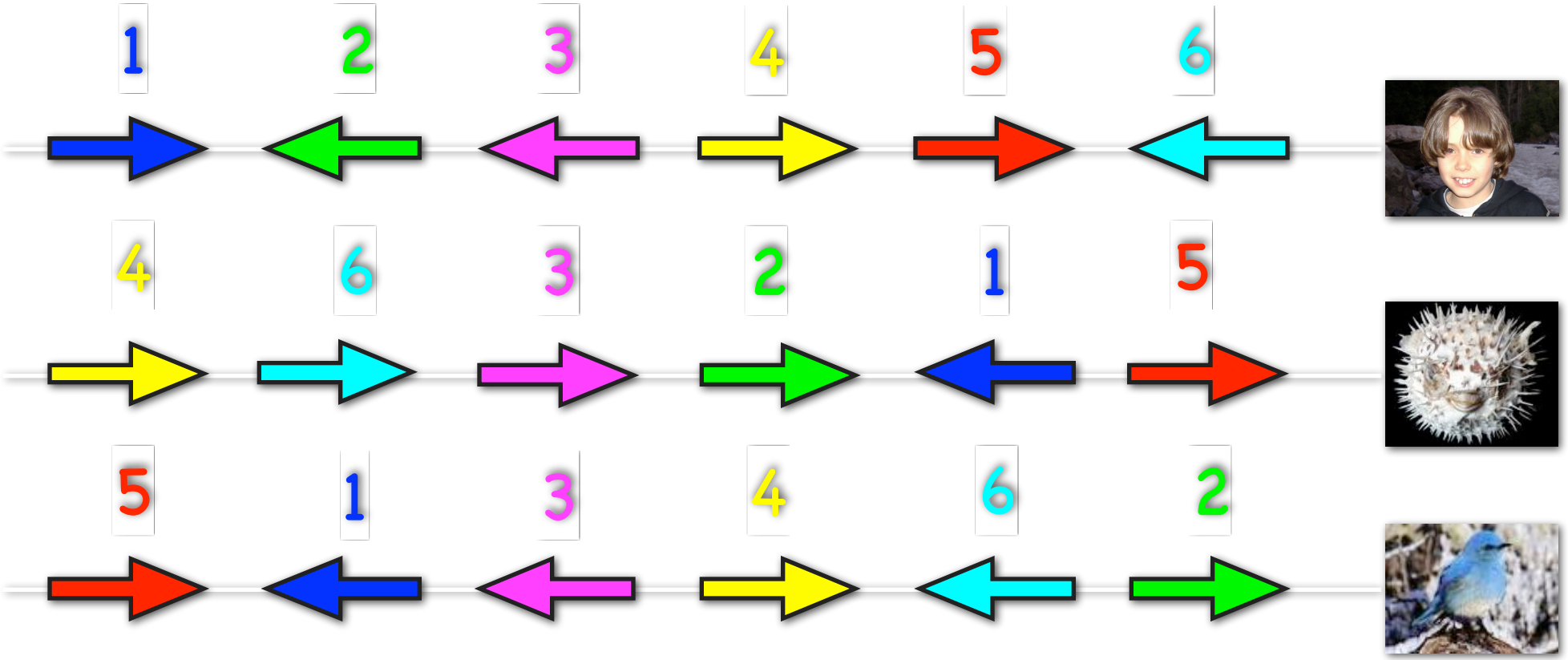
b) Fission et fusion



Wiki: Released public domain by Evercat. Diagram of ancestral fusion forming Human Chromosome 2.

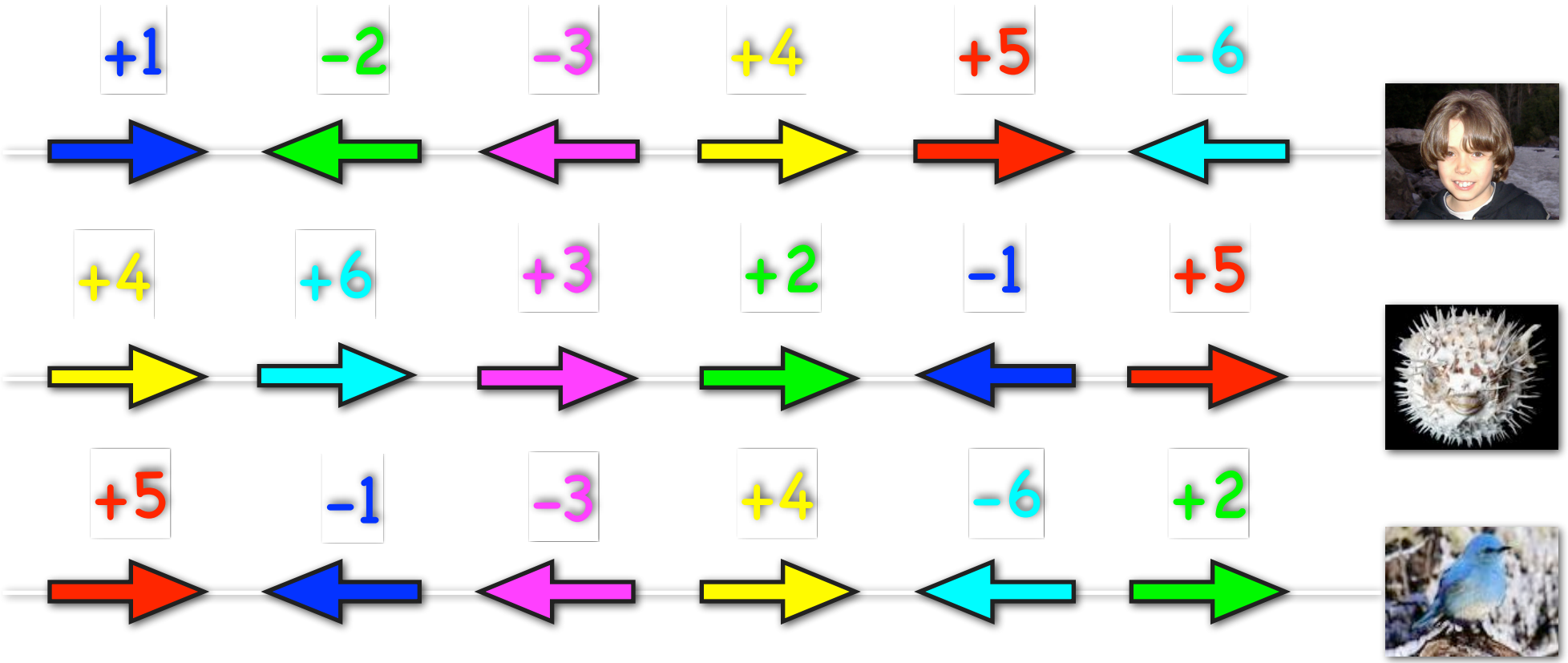
Ordre des gènes dans un génome

Représentation 1: Permutations



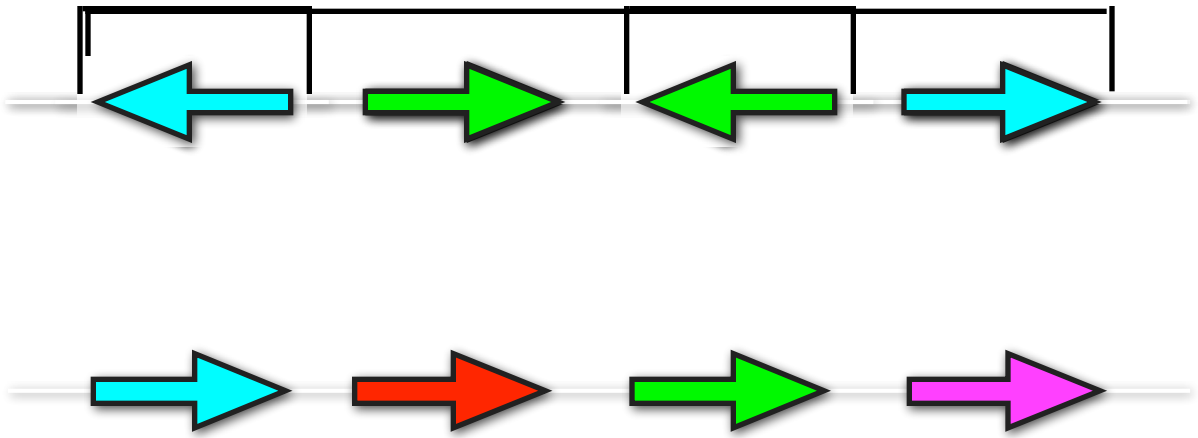
Ordre des gènes dans un génome

Représentation 2: Permutations signée

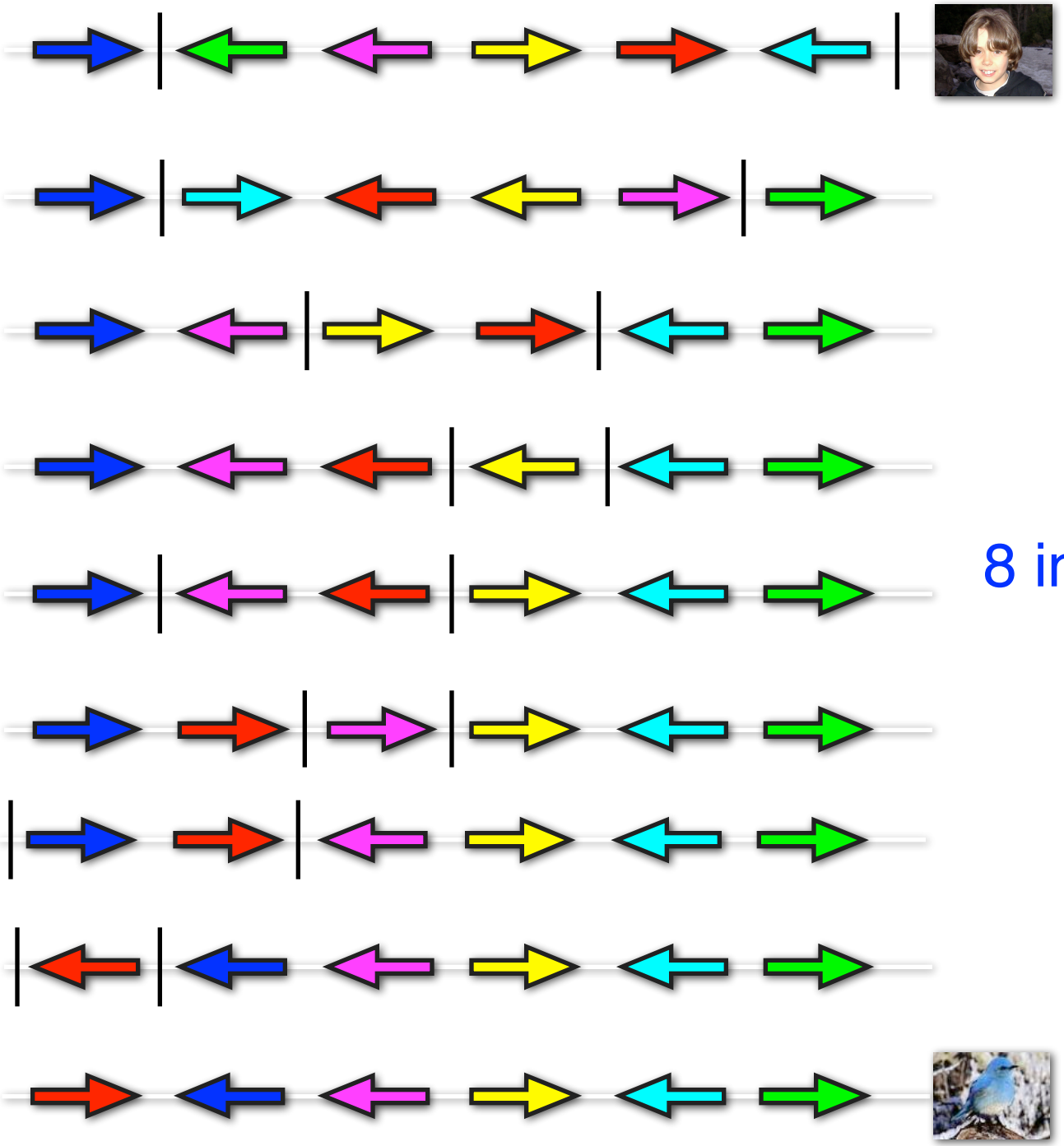


Réarrangement par inversions

Problème: Étant donné deux segments G et H de génomes différents contenant les mêmes gènes mais dans un ordre différent, trouver le nombre minimal d'inversions pour passer de G à H.



nombres d'inversions: 2



8 inversions

Les premiers articles sur le sujet

- Kececioglu et Sankoff, 1993[1]: Première heuristique dans le cas des gènes non signés

- Caprara, 1999[2]: Prouve que le problème est NP-difficile pour le cas des gènes non signés

- Hannenhalli et Pevzner, 1995[3]: Premier algorithme polynomial pour le cas des gènes signés

1 J. Kececioglu and D. Sankoff, *Exact and approximation algorithms for the inversion distance between two chromosomes*, **Combinatorial Pattern Matching.** LNCS 684, p. 87-105, 1993.

2 A. Caprara, *Sorting Permutations by Reversals and Eulerian Cycle Decompositions*, ***SIAM Journal on Discrete Mathematics*** 12, pp. 91-110, 1999.

3 S. Hannenhalli and P. Pevzner, *Transforming Cabbage into Turnip (polynomial algorithm for sorting signed permutations by reversals),* **Journal of the ACM**, pp. 178-189, 1995.

Réarrangement par inversions - formalisation du problème dans le cas des permutations

Une séquence de n gènes est représentée par une permutation π, où une permutation est une bijection de l'ensemble des n premiers entiers vers lui-même

Une inversion, dénotée $\rho(i, j)$, renverse l'ordre des éléments de la position i à la position j dans π

$$\pi = \pi_1 \pi_2 \ldots \pi_{i-1} \pi_i \pi_{i+1} \ldots \pi_{j-1} \pi_j \pi_{j+1} \ldots \pi_n$$

$$\downarrow \rho(i, j)$$

$$\pi = \pi_1 \pi_2 \ldots \pi_{i-1} \pi_j \pi_{j-1} \ldots \pi_{i+1} \pi_i \pi_{j+1} \ldots \pi_n$$

Distance d'inversions

But: Étant donné deux permutations de {1,2,…,n}, trouver la plus petite suite d'inversions permettant de transformer une permutation en l'autre.

Entrées: Les deux permutations π et σ

Sortie: Une suite d'inversions $\rho_1, \rho_2, \ldots, \rho_t$ qui transforme π en σ et telle que t est minimal.

Ce t minimal est appelé la distance d'inversions entre π et σ et est dénoté $d(\pi, \sigma)$

Trier par inversions

On va assumer qu'une des permutations considérées est la permutation identité $\mathrm{Id} = 1\ 2\ 3\ \ldots\ n$.

Le problème devient alors un problème de tri.

Trier par inversions: Étant donnée une permutation π, trouver la plus petite suite d'inversions permettant de la trier i.e de la transformer en la permutation identité

Entrées: Une permutation π

Sortie: Une suite d'inversions $\rho_1, \rho_2, \ldots, \rho_t$ qui transforme π en Id et telle que t est minimal. Ce t minimal est appelé la "distance d'inversions" de π, dénoté $d(\pi)$

Trier par inversions - Exemple

Entrée: $\pi = 2\ 4\ 3\ 5\ 8\ 7\ 6\ 1$

On peut trier cette permutation en 4 étapes avec les inversions suivantes:

$$\pi = 2\ \underline{4\ 3}\ 5\ 8\ 7\ 6\ 1$$

$$\underline{2\ 3\ 4\ 5}\ 8\ 7\ 6\ 1$$

$$5\ 4\ 3\ 2\ \underline{8\ 7\ 6\ 1}$$

$$\underline{5\ 4\ 3\ 2\ 1}\ 6\ 7\ 8$$

$$1\ 2\ 3\ 4\ 5\ 6\ 7\ 8$$

modifié de www.bioalgorithms.info

Trier par inversions - Exemple

Peut-on faire mieux?

Est-ce possible de trier cette permutation en 3 étapes? En 2 étapes?

Comment savoir?

Détour gourmand

Un problème un peu plus spécifique et qui date du début des années 1970 est le problème suivant:

> Un chef prépare une pile de crêpes mais ne fait pas attention à l'ordre dans lequel il les empile.

Le serveur veut rapidement réarranger les crêpes par ordre de grandeur de sorte que la plus grande soit en bas de la pile.

Il le fera en retournant plusieurs fois un certain nombre de crêpes, toujours en partant du haut de la pile

http://plus.maths.org/content/pancakes-mice-and-men

Problème de retournement de crêpes

But: Étant donné une pile de n crêpes, quel est le nombre minimal de retournements requis pour réarranger la pile en une pile parfaite

Entrées: Une pile de n crêpes

Sortie: Une suite minimale de retournements pour transformer cette pile en une pile parfaite i.e. une pile ordonnée de telle sorte que la plus grande crêpe soit tout en bas et la plus petite au dessus.

Formalisation mathématique??

Problème de retournement de crêpes - formalisation mathématique

Pour formaliser ce problème de façon mathématique, étiquetons les n crêpes de sorte que la plus petite est l'étiquette 1 et la plus grande, l'étiquette n

La pile de crêpe étiquetée peut donc être représentée par une permutation des n premiers entiers

Le retournement d'un certain nombre de crêpe à partir du haut de la pile peut être vu comme une inversion spéciale qui inclut toujours le premier élément d'une permutation, nommée inversion préfixe.

Le problème de trier une pile de crêpe peut donc être reformulé comme un problème de tri d'une permutation par inversions préfixes.

Trier par inversions préfixes

Trier par inversions: Étant donnée une permutation π, trouver la plus petite suite d'inversions préfixes permettant de la trier i.e de la transformer en la permutation identité

Entrées: Une permutation π

Sortie: Une suite d'inversions préfixes $\rho_1, \rho_2, \ldots, \rho_t$ qui transforme π en Id et telle que t est minimal.

Trier par inversions préfixes

Approche vorace: À chaque étape sélectionner le plus gros élément pas encore traité et l'amener à sa bonne position en utilisant au plus 2 inversions préfixes.

Approche vorace - pire cas: $2n - 2$ inversions préfixes

Bill Gates et Christos Papadimitriou*ont montré en 1979 que ce problème peut être résolu en au plus

$$\frac{5(n+1)}{3} \quad \text{inversions préfixes}$$

* Gates, W. and Papadimitriou, C., *Bounds for Sorting by Prefix Reversal,* **Discrete Mathematics**, 27, pp.47-47, 1979

Retour sur le tri par inversions

On va dénoter $prefix(\pi)$ le plus long préfixe déjà trié d'une permutation π

Exemple: Si $\pi = 1 \quad 2 \quad 6 \quad 3 \quad 5 \quad 4$, alors $prefix(\pi) = 2$

Idée vorace: Augmenter la valeur de $prefix(\pi)$ à chaque étape

Algorithmes approximatifs

Algorithme approximatif: Algorithme qui trouve rapidement une solution approximative à un problème plutôt qu'une solution optimale.

On définit le ratio d'approximation de l'algorithme A sur l'entrée π comme

$$\frac{A(\pi)}{\mathrm{OPT}(\pi)}$$

où $A(\pi) =$ solution produite par l'algorithme A

$\mathrm{OPT}(\pi) =$ solution optimale du problème

Algorithmes approximatifs - guarantie de performance

Le ratio d'approximation (guarantie de performance) d'un algorithme A est le ratio d'approximation maximal sur toutes les entrées de taille n, i.e.

$$\max_{|\pi|=n} \frac{A(\pi)}{\mathrm{OPT}(\pi)}$$

Question: Quel est le meilleur ratio d'approximation qu'on peut obtenir pour le tri par inversions?

Adjacences et points de cassure

Soit $\pi = \pi_1 \pi_2 \ldots \pi_n$ une permutation. Une paire d'éléments π_i et π_{i+1} est appelée une adjacente si

$$\pi_{i+1} = \pi_i + 1 \quad \text{ou} \quad \pi_i - 1$$

Toutes les autres paires de points sont appelées des points de cassure

Étendre les permutations

Pour "encrer" une permutation on ajoute un élément au début, $\pi_0 = 0$, et un élément à la fin $\pi_{n+1} = n + 1$

Tri par inversions = élimination de point de cassures

La permutation identité $\mathrm{Id} = 1\ 2\ 3\ \ldots\ n$ ne comporte aucun point de cassure.

Une idée vorace pour trier par inversions est donc d'essayer de diminuer le nombre de points de cassure d'une permutation

On va dénoter $b(\pi)$ le nombre de points de cassure dans l'extension de π

Tri par inversions = élimination de point de cassures

Combien de points de cassure peut-on éliminer avec une inversion?

2

Cela implique que $d(\pi) \geq \frac{b(\pi)}{2}$

On peut donc construire un nouvel algorithme vorace avec cette idée de réduire les points de cassure.

Quelques définitions

On appelle bande un intervalle entre deux points de cassure.

bande décroissante: une bande d'éléments en ordre décroissant

bande croissante: une bande d'éléments en ordre croissant

Pour les besoins de la suite, une bande d'un seul élément, autre que 0 et n+1, sera déclarée décroissante

Réduire le nombre de points de cassure

Théorème 1: Si une permutation contient au moins une bande décroissante, alors il existe une inversion qui va réduire le nombre de points de cassure

On va utiliser ce théorème pour transformer notre algorithme glouton ayant une garantie d'arrêt

exemple au tableau

Le cas sans bande décroissante

S'il n'y a pas de bande décroissante, il est possible qu'aucune inversion ne fasse descende le nombre de point de cassures

Par contre, en inversant une bande croissante, on va créer pour la prochaine étape une bande décroissante

Le théorème 1 garantit alors une réduction du nombre de point de cassures au prochain tour

Tout cela nous donne un nouvel algorithme vorace