08/03/2021 (présentiel)

EA4 TD5 : Permutations, tri par fusion

Exercice 1

1) $\sigma = (1\ 5)(2\ 9\ 4)(3\ 7\ 10\ 6)$ /

$\tau = (2\ 5)(3\ 8)(6\ 7)$ /

[On ne note pas les points fixes dans la notation cyclique]

2) $\sigma\tau = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 9 & 6 & 5 & 1 & 8 & 10 & 3 & 4 & 7 \end{pmatrix}$

$= (1\ 5\ 9\ 4\ 2)(3\ 8\ 7)(6\ 10)$

$\tau\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 1 & 8 & 2 & 9 & 10 & 3 & 7 & 4 & 6 \end{pmatrix}$

$= (1\ 2\ 9\ 4\ 5)(3\ 6\ 8)(7\ 10)$

Autre méthode sur le cycle :

$\sigma\tau = (1\ 5)(2\ 9\ 4)(3\ 7\ 10\ 6).(2\ 5)(3\ 8)(6\ 7)$

$= (2\ 9\ 4)(1\ 5)(2\ 5)(3\ 7\ 10\ 6)(3\ 8)(6\ 7)$ ← on regroupe les cycles en fct° de leur support (= les éléments en commun).

$= (2\ 1\ 5\ 9\ 4)(3\ 8\ 7)(6\ 10)$

⚠ on ne peut PAS commuter des cycles qui ont un élément (support) commun.

3) $\sigma^{-1} = (5\ 1)(4\ 9\ 2)(6\ 10\ 7\ 3)$

$\tau^{-1} = (5\ 2)(8\ 3)(7\ 6)$

$(\sigma\tau)^{-1} = (5\ 4\ 9\ 2\ 1)(8\ 6\ 3)(10\ 7)$

normalement, il faudrait inverser l'ordre des cycles : $(A\,B\,C)^{-1} = C^{-1}B^{-1}A^{-1}$

↳ mais ici les cycles ont un support disjoint (pas d'élément en commun) donc ils commutent.

pour chaque permutation, on regarde, pour chaque élément en partant de la fin.

4)

	1	2	3	4	5	6	7	8	9	10
	5	2	3	4	1	6	7	8	9	10
	5	9	3	4	1	6	7	8	2	10
	5	9	3	2	1	6	7	8	4	10
	5	9	3	2	1	3	7	8	4	6
	5	9	3	2	1	6	10	8	4	3
$\sigma \leftarrow$	5	9	7	2	1	6	3	8	4	10

$\sigma = (2\ 5)(3\ 8)(6\ 7)$

$\sigma^{-1} = (6\ 7)(3\ 8)(2\ 5)$

$\sigma\tau = (1\ 5\ 9\ 4\ 2)(3\ 8\ 7)(6\ 10)$

$(\sigma\tau)^{-1} = [(2\ 4)(2\ 9)(2\ 5)(2\ 1)\ (7\ 8)(7\ 3)(6\ 10)]^{-1}$

$(\sigma\tau)^{-1} = (6\ 10)(7\ 3)(7\ 8)(2\ 1)(2\ 5)(2\ 9)(2\ 4)$

$(\sigma\tau)^{-1} = (1\ 2\ 4\ 9\ 5)(3\ 7\ 8)(6\ 10)$

Plus généralement :

$(A\ B\ C \ldots N)^{-1} = (N\ A)(N\ B)(N\ C)\ldots$

ex : $(2\ 9\ 4)^{-1} = (4\ 2)(4\ 9)$

Donc $\sigma = (3\ 7)(10\ 3)(6\ 3)(2\ 4)(9\ 2)(1\ 5)$

[Il faut l'écrire à l'envers car on commence à lire l'expression par la droite]

puis regrouper les cycles comme au 2)

Exercice 2

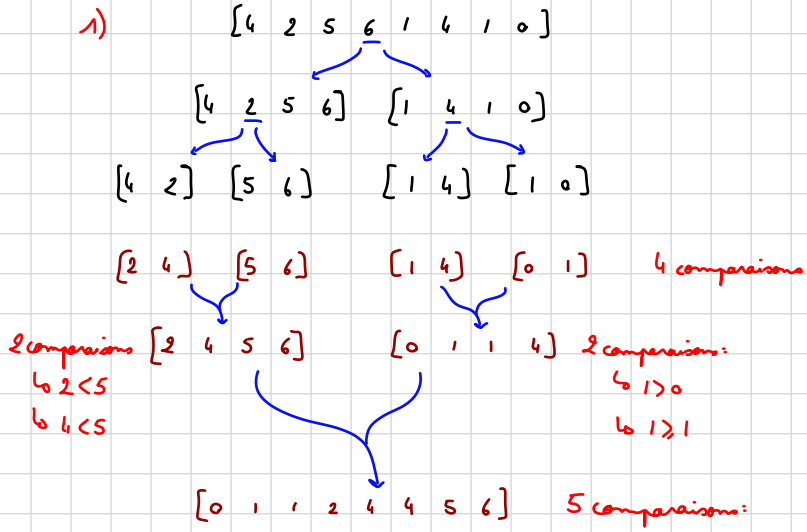
1)
```
def estPerm(T):
    S = [-1] * T.length();
    for i in T:
        if S[i] != -1:
            return False
        S[i] = 1
    for i in S:
        if S[i] == -1:
            return False
    return True
```

2)
```
def inversePerm(T):
    if ! estPerm(T):          ← o(n)
        return None
    S = [-1] * T.length()
    for i in T:               ← n tours de boucle ⇒ o(n)
        S[T[i]] = i
    return S
```

3)
```
def composePerm(T, S):
    if (! estPerm(T) || ! estPerm(S) || T.length() != S.length():
        return None
    R = [-1] * T.length()
    for i in T:
        R[i] = S[T[i]]
    return R
```

Exercice 3

1)



Donc 13 comparaisons en tout.

2)

```
def fusionItérative(T, S):
    R = [0] * (T.length + S.length)
    i = 0
    j = 0
    for k in range(0, R.length):
        if i == T.length :
            R[k] = S[j]
            j++
        else if j == S.length :
            R[k] = T[i]
            i++
        else :
            if T[i] < S[j] :
                R[k] = T[i]
                i++
            else
                R[k] = S[j]
                j++
    return R
```