R1.01 - Initiation au développement

Date: 14 Nov 2023

Sujet: mélanger + inverser un les tableaux

Instructeur: Joseph AZAR - joseph.azar@univ-fcomte.fr

Problème:

L'algorithme doit simuler un jeu appelé *reversi* Le jeu commence par afficher les chiffres de 1 à 9 dans le désordre. Le joueur donne un entier compris entre 1 et 9 qui indique le nombres de chiffres à inverser à partir de la gauche. La nouvelle configuration du jeu est affichée. Par exemple pour la configuration :

5 4 6 2 1 7 9 8 3

si le joueur donne l'entier 5, la nouvelle configuration sera:

1 2 6 4 5 7 9 8 3

Le jeu s'arrête quand on atteint une configuration où les chiffres sont placés par ordre croissant. L'objectif du joueur est de réaliser ce but avec le moins de tours possible. L'algorithme permettra d'imprimer en combien de tours le joueurs a réalisé le tri et lui permettra de recommencer le jeu sur la même configuration initiale autant de fois qu'il le désire. La configuration initiale est supposée fournie par une fonction:

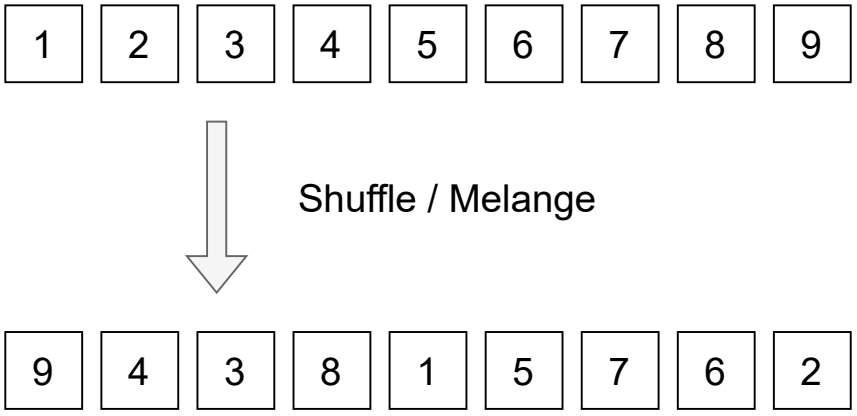
```
fonction initTab(out Tab : entier tableau, in taille : entier) : vide
```

dont on ne décrira pas l'algorithme.

Un premier pseudo-code

```
Debut

    tab : <----- tableau entier avec taille = 9

    nbTours : <----- entier = 0

     i : <----- index entier (saisie par le joueur)

     tab : <------ initTab(tab)

     Tant que 1=1 <------- Le jeu commence ici

                afficherTab(tab)

                Repeter
                      print("Veuillez saisir une valeur entre 2 et 9. 99 pour quitter")

                      i : <----- saisir index

               Tant que i n'est pas entre 2 et 9 ou 99

               Si i == 99
                    break

               nbTours : <----- nbTours + 1

               tab : <------ inverseTab(tab,i)

                Si estTrie(tab)

                      print("Vous avez resolu ce jeu en " + nbTours + " tours")

                      break

Fin
```

initTab



Algorithme

Pour i allant de 0 à N-1

tab[i] ⟵ i+1

melangerTab(tab)

melangerTab

Approche 3 :
Le tableau est parcouru de gauche à droite, et la valeur de chaque case est échangée avec celle d'une autre case se trouvant après elle dans le tableau. Cette autre case est choisie de manière aléatoire. On utilise pour cela une fonction qui retourne un nombre aléatoire.

Algorithme

Pour i allant de 0 a N-2

    j ⟵ nombreAleatoire(i,N) [i,N[

    echanger(tab,i,j)

1	2	3	4	i = 0	j = 1

2	1	3	4	i = 1	j = 3

2	4	3	1	i = 2	j = 3

2	4	1	3

nombreAleatoire (min, max)

Algorithme

intervalle ⟵ (max - min) (On ajoute 1 pour que max soit inclus)

j ⟵ (Math.random x intervalle) + min

Exemple:

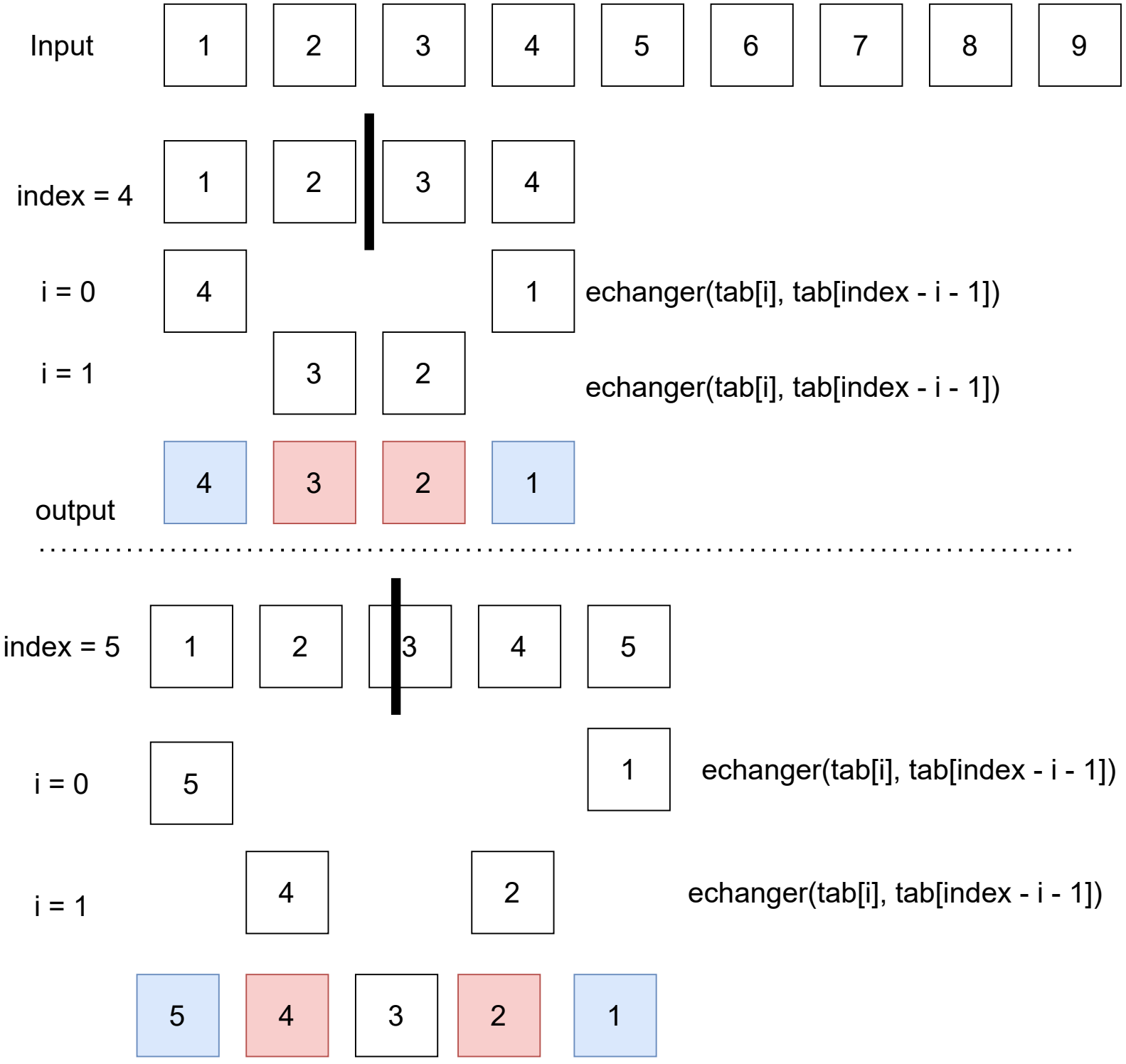
min = 0
max = 2
intervalle = 2-0 = 2
Math.random = 0.9
j = 0.9 x 2 + 0
j = int(1.8) = 1

min = 0
max = 2
intervalle = 2-0 = 2
Math.random = 0.45
j = 0.45 x 2 + 0
j = int(0.9) = 0

min = 2
max = 4
intervalle = 4-2 = 2
Math.random = 0.9
j = 0.9 x 2 + 2
j = int(3.8) = 3

min = 2
max = 4
intervalle = 4-2 = 2
Math.random = 0.45
j = 0.45 x 2 + 2
j = int(2.9) = 2

inverse (exemple)



inverse

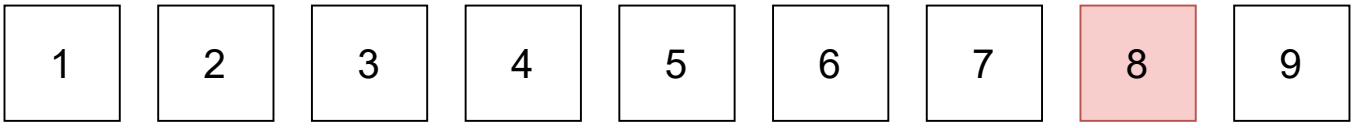
Algorithme

index (entrée)

Pour i allant de 0 à int(index/2)

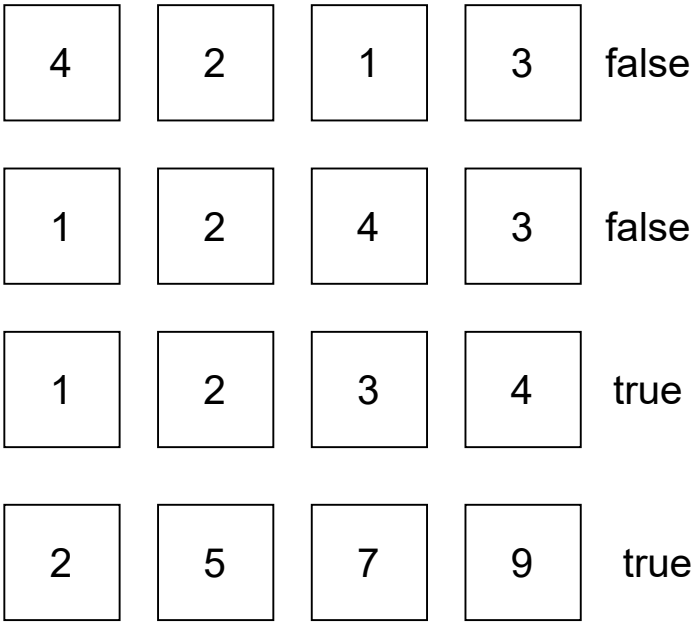
    echanger(tab[i], tab[index-i-1])

estTrie

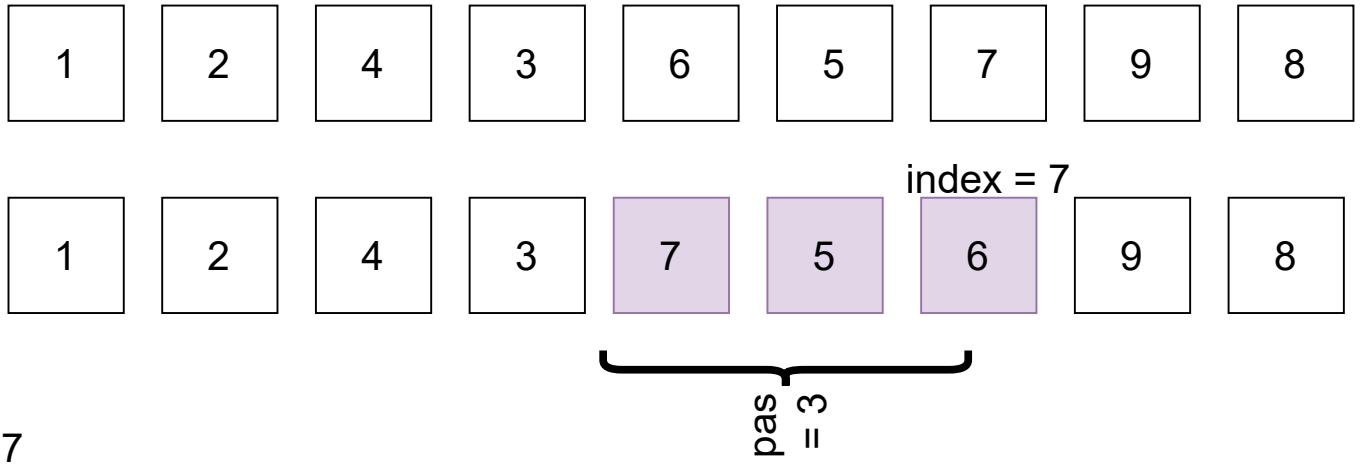


Algorithme

```
Pour i allant de 0 à N-2
    Si tab[i] > tab[i+1]
        retourner false
retourner true
```



Améliorer l'algorithme actuel



index = 7

pas = 3