Trier des cartes

Christophe Viroulaud

Première - NSI

cartes

nuellement

Trier des cartes

isposer au tri Ionnées ar sélection

elementation minaison rection

plexité

r insertion émentation

de terminaison

de correction exité

Sommaire

- 1. Problématique

Implémentation

Terminaison

Correction

Complexité Implémentation

Preuve de terminaison

Preuve de correction Complexité

Problématique

Trier des cartes

Trier un jeu de cartes est une opération qui trouve des applications en informatique.

Existe-t-il plusieurs méthodes pour trier des données?

Sommaire

- 2. Trier des cartes manuellement

Implémentation Terminaison

Correction

Complexité

Implémentation Preuve de terminaison

Preuve de correction Complexité

Trier des cartes

manuellement

Trier des cartes



 ${\rm Figure} \ 1 - {\sf Cartes} \ {\sf m\'elang\'ees}$

Activité 1 :

- Prendre le paquet de cartes mélangées et les étaler
 cor la toble
- 2. Trier les cartes.
- Formaliser la méthode utilisée sous forme d'un algorithme.

Activité 1 :

- 1. Prendre le paquet de cartes mélangées et les étaler sur la table.
- 2. Trier les cartes.
- 3. Formaliser la méthode utilisée sous forme d'un algorithme.

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

par sélection

orrection

complexité

plémentation

ve de terminaison ve de correction

lexité

Différentes méthodes

► Tri par sélection en place

Trier des cartes

- ► Tri par sélection en place
- ► Tri par sélection dans un nouveau tableau
- ► Tri par insertion en place
- ► Tri par insertion dans un nouveau tableau

Pour chaque carte du tas
Trouver la plus petite carte dans la partie non tri
ée.

Échanger cette carte avec la première de la partie non triée.

Code 1 – Tri par sélection (en place)

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

manuellement

e données Tri par sélection

Terminaison

Complexité

Implémentation

reuve de terminaison

ıplexité



Tri par sélection en place

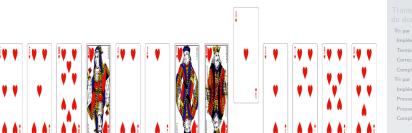


FIGURE 2 – Sélectionne la plus petite du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

Tri par sélection Implémentation

Terminaison Correction

Complexité

Tri par insertion

Implémentation

reuve de terminaison

mplexité



Tri par sélection en place

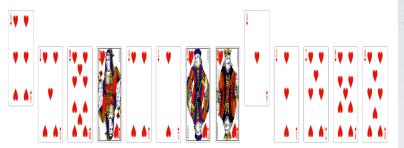


FIGURE 3 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tr

ri par sélection

Terminaison Correction Complexité

Tri par insertion

Implémentation
Preuve de terminaisor

reuve de terminaison reuve de correction

mplexité



Tri par sélection en place

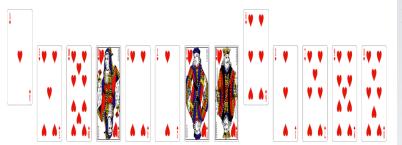


FIGURE 4 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tr de données

Tri par sélection Implémentation Terminaison

Correction Complexité

Tri par insertion

Implémentation
Preuve de terminaisor

Preuve de terminaison Preuve de correction

Tri par sélection en place

Retour menu

manuellement

FIGURE 5 – La carte est à sa place

Retour menu



Tri par sélection en place



FIGURE 6 – Sélectionne la plus petite du tas non trié

Retour menu

Trier des cartes

Trier des cartes manuellement



Tri par sélection en place

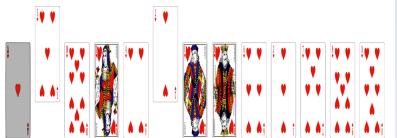


FIGURE 7 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

ri par sélection

Terminaison Correction

Complexité Tri par insertion

Implémentation

Preuve de terminaison

reuve de correction Complexité

Trier des cartes Trier des cartes manuellement

└─Tri par sélection en place



Tri par sélection en place

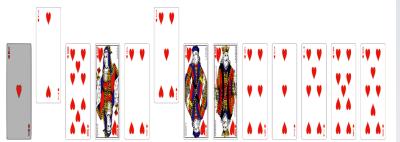


FIGURE 8 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

roblématique

Trier des cartes manuellement

Transposer au tri

Tri par sélection Implémentation

Correction

Complexite Tri par insertion

Implémentation

reuve de terminaison

mplexité

Retour menu

Tri par sélection en place

Retour menu

FIGURE 9 – La carte est à sa place

Pour chaque carte du tas

Trouver la plus petite carte du tableau non trié.

La placer à la fin du tableau trié.

Code 2 – Tri par sélection dans un nouveau tableau

Retour menu

Trier des cartes

Trier des cartes manuellement

Retour menu

Tri par sélection - nouveau tableau

Trier des cartes

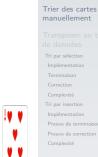
Trier des cartes manuellement

FIGURE 10 – Trouve la plus petite du tas non trié

Retour menu

Tri par sélection - nouveau tableau

Tri par sélection - nouveau tableau



Trier des cartes

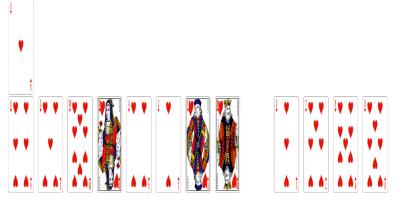


FIGURE 11 – La place à la fin du tableau trié

Retour menu

└─Tri par sélection - nouveau tableau



Tri par sélection - nouveau tableau

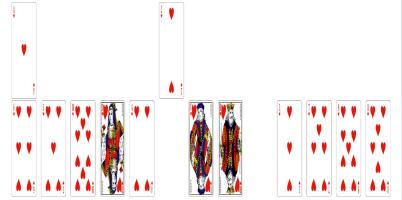


FIGURE 12 – Trouve la plus petite du tas non trié

Retour menu

Trier des cartes

oblématique

Trier des cartes manuellement

Transposer au t

Tri par sélection
Implémentation
Terminaison
Correction

Tri par insertion

Implémentation
Preuve de terminaison
Preuve de correction

Preuve de correction Complexité

Tri par sélection - nouveau tableau

Retour menu

Tri par sélection - nouveau tableau

Trier des cartes

Trier des cartes manuellement

FIGURE 13 – La place à la fin du tableau trié

Retour menu

Pour chaque carte du tas
 Mémoriser la carte en cours
 Décaler vers la droite toutes les cartes précédentes, supérieures à la carte en cours.

Insérer la carte en cours dans l'espace vide.

Code 3 – Tri par insertion (en place)

Retour menu

4

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au t

ri par sélection Implémentation

orrection

ar insertion

uve de terminaison

reuve de correction

mplexité

Tri par insertion en place

Tri par insertion en place

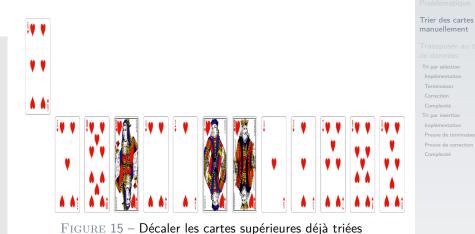
Trier des cartes

Trier des cartes manuellement

FIGURE 14 – Mémoriser la première carte dans le tas non trié

Retour menu

Tri par insertion en place



Retour menu

Trier des cartes

Retour menu

2021-05-27 ☐Tri par insertion en place la carte est dans le tas trié

FIGURE 16 – Replacer la carte dans l'espace

☐Tri par insertion en place

la carte est dans le tas trié



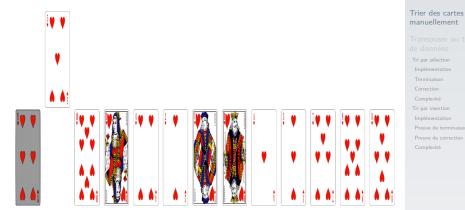


FIGURE 17 – Mémoriser la première carte dans le tas non trié

Retour menu

Trier des cartes

☐Tri par insertion en place

lci pas de carte triée encore

Tri par insertion en place

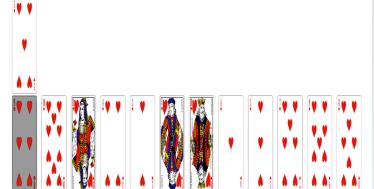


FIGURE 18 – Décaler les cartes supérieures déjà triées

Retour menu

Trier des cartes

Trier des cartes manuellement

☐Tri par insertion en place

la carte est dans le tas trié

Tri par insertion en place



FIGURE 19 – Replacer la carte dans l'espace

Retour menu

Trier des cartes

oblématique

Trier des cartes manuellement

ransposer au tri

ri par sélection

Terminaison Correction

Complexité

Tri par insertion

Implémentation

Preuve de terminaiso

euve de correction

plexité

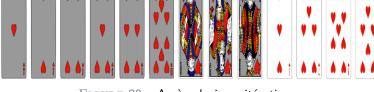


FIGURE 20 – Après plusieurs itérations

Retour menu

Tri par insertion en place

Retour menu

Form 27 - Managar is present som dan be in one and

Tri par insertion en place

la carte est dans le tas trié

Tri par insertion en place

FIGURE 21 – Mémoriser la première carte dans le tas non trié

Retour menu

Trier des cartes

manuellement

Trier des cartes

ansposer au tr

Implémentation Terminaison Correction

Complexité Tri par insertion

Tri par insertion Implémentation

Preuve de terminai

euve de correction mplexité



Trier des cartes

Trier des cartes

manuellement

FIGURE 22 – Décaler les cartes supérieures déjà triées

Retour menu

Retour menu

Tri par insertion en place

Tri par insertion en place

Trier des cartes

manuellement

Trier des cartes

FIGURE 23 – Insérer la carte dans l'espace

Retour menu

Pour chaque carte du tas

Prendre la première carte du tableau non trié.

Dans le tableau trié, décaler vers la droite toutes les

cartes plus grandes.

Insérer la carte dans le tableau trié.

Code 4 – Tri par insertion dans un nouveau tableau

Retour menu

4

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tr

e données Fri par sélection

Terminaison

Correction

par insertion

nplémentation

reuve de terminaison

mplexité

Tri par insertion - nouveau tableau

Tri par insertion - nouveau tableau

Trier des cartes

Trier des cartes manuellement

FIGURE 24 – Prendre la première carte non triée

Retour menu

Tri par insertion – nouveau tableau

Tri par insertion - nouveau tableau

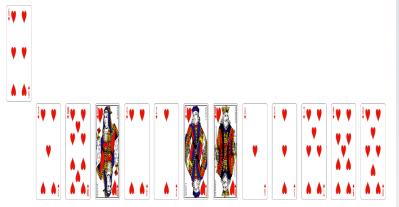


FIGURE 25 – Décaler les cartes supérieures du tableau trié

Retour menu

Trier des cartes

oblématique

Trier des cartes manuellement

nanuellement Fransposer au ti

ri par sélection mplémentation Ferminaison Correction

Complexité
Tri par insertion
Implémentation

Implémentation

Preuve de terminaiso

Preuve de correction

Preuve de correction Complexité

Tri par insertion - nouveau tableau

FIGURE 26 - Insérer la carte dans le tableau triée Retour menu

Tri par insertion - nouveau tableau

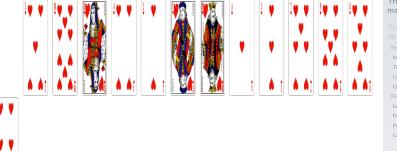


FIGURE 26 – Insérer la carte dans le tableau triée

Retour menu

Trier des cartes manuellement

Trier des cartes



Tri par insertion - nouveau tableau

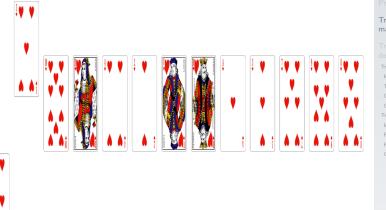


FIGURE 27 – Prendre la première carte non triée

Trier des cartes

Trier des cartes

manuellement

37 / 76

Trier des cartes manuellement

☐Tri par insertion - nouveau tableau



Tri par insertion - nouveau tableau

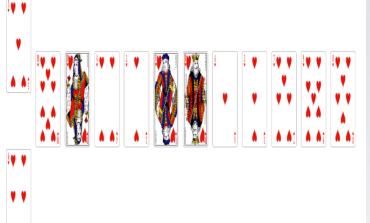


FIGURE 28 – Décaler les cartes supérieures du tableau trié

Retour menu

ématique

Trier des cartes

Trier des cartes manuellement

ansposer au tr données

Terminaison Correction

Complexité Tri par insertio

Tri par insertion Implémentation

Preuve de termir

Preuve de corre

euve de corrections omplexité

Tri par insertion - nouveau tableau

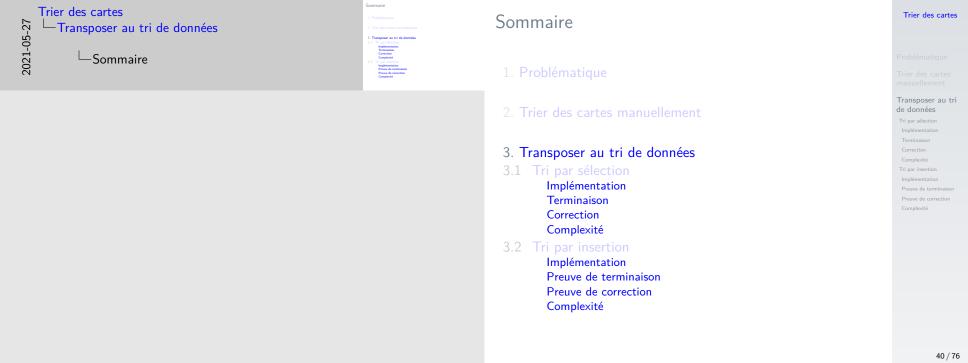
Retour menu

FIGURE 29 - Insérer la carte dans le tableau triée

manuellement

FIGURE 29 – Insérer la carte dans le tableau triée

Retour menu

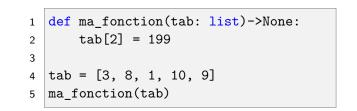


Trier des cartes Transposer au tri de données

- Quand on pass on tableace are agreement à une fonction cette dennées neurophie la tableace original.

 | del ma_functions(tabs: liste) 950mm; |
 | del ma_functions | 100mm; |
 | del ma_functions | 10
- 1. Il faut avoir conscience que les données d'origine sont modifiées. Il ne sert à rien que la fonction renvoie le tableau.
- 2. on va implémenter tri en place; tri dans nouveau tableau dans exercices

Quand on passe un tableau en argument à une fonction, cette dernière manipule le tableau original.



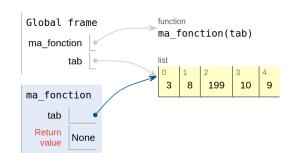


FIGURE 30 – La fonction modifie le tableau original

Problématique

Trier des cartes

Trier des cartes

Transposer au tri

i par sélection nplémentation erminaison

omplexité i par insertion nplémentation

euve de terminai

Rappel de l'algorithme

Rappel de l'algorithme

Pour chaque carte du tas

Trouver la plus petite carte dans la partie non triée.

Échanger cette carte avec la première de la partie non triée.

Code 5 – Tri par sélection (en place)

Trier des cartes

Problèmatique

Trier des cartes manuellement

> ansposer au tri données

Tri par sélection Implémentation

Terminaison Correction

Fri par insertion

olémentation uve de terminaison

ve de correction

Activité 2 :

1. Écrire la fonction treuver_mini(tab : Est) --- int
qui renvoire l'indice du plus patit élément de zab.

Tri par sélection - question préparatoire

Tri par sélection - question préparatoire

Activité 2 :

1. Écrire la fonction **trouver_mini(tab : list)** → **int** qui renvoie l'indice du plus petit élément de *tab*.

Trier des cartes

Problématique

Trier des cartes manuellement

Fransposer au tri le données

ar sélection

Implémentation

rrection

Complexité

nplémentation

euve de terminaison

ve de correction

Correction

Correction

```
def trouver_mini(tab: list) -> int:
    """

Trouve l'indice du plus petit élément
    """

i_mini = 0

for j in range(1, len(tab)):
    if tab[j] < tab[i_mini]:
        i_mini = j

return i_mini</pre>
```

Trier des cartes

Problématique

rier des cartes

ransposer au tri e données

par sélection

Implémentation

Correction

omplexité

par insertion

reuve de terminaison

uve de terminaison uve de correction Fri par sélection - Implémentation

- l'indice du plus petit élément de tab, compris entre l'indice i depart et la fin du tableau. La signature de la fonction deviendra trouver_mini(tab : list
- Écrire la fonction echanger(tab : list, i : int, j
- int) -> None qui échange les éléments d'indice
- Écrire la fonction tri selection(tab : list) ->

None qui effectue un tri par sélection sur tab.

Tri par sélection - Implémentation

Activité 2 :

2. Adapter la fonction précédente pour renvoyer l'indice du plus petit élément de tab, compris entre l'indice i_depart et la fin du tableau. La signature de la fonction deviendra trouver_mini(tab : list, i_depart : int) \rightarrow int.

- 3. Écrire la fonction **echanger(tab : list, i : int, j :** int) \rightarrow None qui échange les éléments d'indice i et i du tableau tab.
- 4. Écrire la fonction **tri_selection(tab : list)** → **None** qui effectue un tri par sélection sur tab.

Trier des cartes

Implémentation

Correction

```
def trouver_mini(tab: list, i_depart: int) -> int:
    """
renvoie l'indice du plus petit élément entre
    i_depart et la fin du tableau
    """
    i_mini = i_depart
    for j in range(i_depart+1, len(tab)):
        if tab[j] < tab[i_mini]:
        i_mini = j
    return i_mini</pre>
```

Trier des cartes

Problématique

Trier des cartes nanuellement

ransposer au tri e données

ri par sélection

Implémentation

rminaison

omplexité

par insertion

iplémentation euve de terminaison

uve de terminaison

def echanger(tab: list, i: int, j: int) -> None:

tab[i], tab[j] = tab[j], tab[i]

inverse les élément d'indices i et j du tableau

Correction

3

Trier des cartes

Implémentation

Correction

Trier des cartes

Problématique

rier des cartes nanuellement

> nsposer au tri données

par sélection

Implémentation

Correction

mplexité

par insertion

npiementation reuse de terminaison

reuve de terminaison

plexité

Tri par sélection - Tester

Tri par sélection - Tester

Activité 2 :

- 5. Construire par compréhension un tableau des entiers de 1 à 13.
- 6. Mélanger le tableau à l'aide de la méthode *shuffle* de la bibliothèque *random*.
- 7. Trier le tableau à l'aide de la fonction *tri_selection*.

Problématique

Trier des cartes

Trier des cartes manuellement

ransposer au tri e données

Tri par sélection

erminaison orrection

Complexité ri par insertion

mplémentation

uve de terminaison

Correction

```
Correction
```

```
cartes = [i for i in range(1, 14)]
  shuffle(cartes)
tri_selection(cartes)
```

Trier des cartes

Implémentation

Preuve de terminaison : variant de boucle

Terminaison La terminaison de la fonction est triviale. Le tri est composé de deux boucles bornées donc qui terminent.

Trier des cartes

51 / 76

Avant chaque ideration de la boucle externe, la partie gau de tableau est triée.

3 8 7 1 5

Code 6 - Avant la première ideration, la partie gauche est vide, donc triée.

Preuve de correction : invariant de boucle

Preuve de correction : invariant de boucle

Avant chaque itération de la boucle externe, la partie gauche du tableau est triée.

3	8	7	1	5
---	---	---	---	---

Code 6 – Avant la première itération, la partie gauche est vide, donc triée.

Trier des cartes

Problematique

Trier des cartes manuellement

données i par sélection

par selection plémentation rminaison

Correction

Complexité

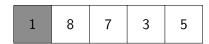
Implémentation

euve de terminaison

ve de correction plexité



Preuve de correction : invariant de boucle



Code 7 – Avant la deuxième itération, la partie gauche est triée.

Trier des cartes

Problematique

Trier des cartes manuellement

nsposer au tri données

par selection nplémentation erminaison

Correction Complexité

Complexité ri par insertion

olémentation

ve de terminaison

e de correction

Preuve de correction : invariant de boucle



Code 8 – Avant la troisième itération, la partie gauche est triée.

Trier des cartes

Correction

Efficacité du tri

La boucle externe effectue n itérations.

 à la première itération de *i*, la boucle de la fonction trouver_mini effectue n-1 itérations.

|--|

 à la deuxième itération de *i*, la boucle de la fonction trouver_mini effectue n-2 itérations.

1 3	7	8	5
-----	---	---	---

...

Trier des cartes

Problématique

Trier des cartes

Transposer au tri

Tri par sélection Implémentation

Correction

Complexité

ri par insertion

nplémentation

euve de terminaison euve de correction

 $\sum_{n=1}^{n-1} k = (n-1) + (n-2) + \cdots + 2 + 1 = \frac{n \cdot (n-1)}{2}$ Le tri par sélection effectue $\frac{n.(n-1)}{2}$ opérations pour Le nombre d'opérations dépend de nº

démo 2 colonnes inversées de 1+2+...+n-1

$$\sum_{k=1}^{n-1} k = (n-1) + (n-2) + \dots + 2 + 1 = \frac{n \cdot (n-1)}{2}$$

À retenir

Le tri par sélection effectue $\frac{n.(n-1)}{2}$ opérations pour ordonner le tableau.

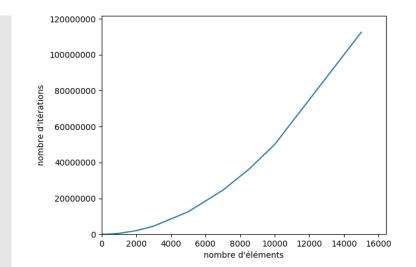
Le nombre d'opérations dépend de n^2 .

Trier des cartes

Evolution du nombre d'itérations

15000 éléments \rightarrow 100 millions d'itérations

Évolution du nombre d'itérations



Trier des cartes

Problématique

Trier des cartes manuellement

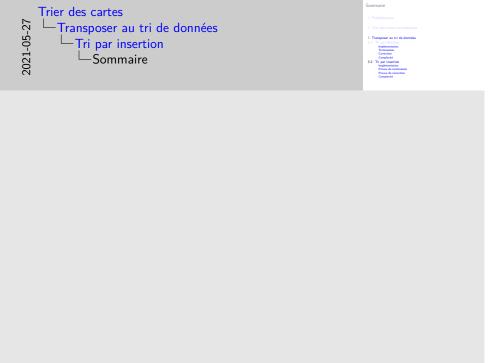
nsposer au tri données

lémentation ninaison

Complexité

plémentation

ve de terminaison



Sommaire

3. Transposer au tri de données

Implémentation Terminaison Correction Complexité 3.2 Tri par insertion Implémentation Preuve de terminaison Preuve de correction

Complexité

Tri par insertion

58 / 76

Trier des cartes

Activité 3:

- Nome en l'approprie pri l'algorithme. Les l'int)
- Nome en l'approprie pri l'algorithme. Les l'inticalités sissiments permittente de contraire les l'inticalités sissiments permittente de contraire les l'inticalités sissiments permittente de contraire les l'inticalités sissiment en propriée produient en couract épositement et pas personne des l'approprie des l'inticalités de l'int

Tri par insertion

Tri par insertion

Activité 3:

- Écrire la fonction tri_insertion(tab : list)
 →None en s'appuyant sur l'algorithme. Les
 indications suivantes permettront de construire les
 trois étapes :
 - Mémoriser : définir une variable en_cours, élément en cours de placement et pos, position actuelle de cet élément.
 - Décaler : utiliser une boucle non bornée pour décaler les éléments vers la droite.
 - ► <u>Insérer</u>: placer l'élément *en_cours* à la nouvelle position *pos*.
- 2. Tester la fonction sur un tableau.

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au tri e données

ri par sélection mplémentation

Correction Complexité

Complexite Tri par insertion

Implémentation
Preuve de terminaison

uve de terminaison uve de correction Rappel de l'algorithme

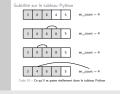
Rappel de l'algorithme

- Pour chaque carte du tas
- Mémoriser la carte en cours
- Décaler vers la droite toutes les cartes précédentes, supérieures à la carte en cours.
- Insérer la carte en cours dans l'espace vide. 4

Code 9 – Tri par insertion (en place)

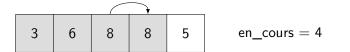
Trier des cartes

Implémentation



Subtilité sur le tableau Python









Code 10 – Ce qu'il se passe réellement dans le tableau Python

Trier des cartes

Problématique

Trier des cartes manuellement

ansposer au tri données par sélection

Terminaison Correction Complexité

Tri par insertion

Implémentation
Preuve de terminaisor

61 / 76

Correction : boucle principale

Correction : boucle principale

Trier des cartes

Problématique

Trier des cartes nanuellement

Transposer au tri de données

par sélection

Terminaison Correction

Complexité
Tri par incortion

Implémentation

reuve de terminaison

ive de correction

Correction: mémoriser

en_cours = tab[i] pos = i Trier des cartes

données
par sélection
plémentation

erminaison

mplexité

Implémentation

ve de terminaison

Correction : décaler

```
\begin{array}{ll} \text{while pos} > 0 \text{ and en\_cours} < \mathsf{tab[pos-1]}: \\ 2 & \mathsf{tab[pos]} = \mathsf{tab[pos-1]} \\ 3 & \mathsf{pos} = \mathsf{pos}{-1} \end{array}
```

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au tri e données

ar sélection émentation

rection

mplexité

Tri par insertion
Implémentation

entation de terminaison

de terminaison de correction

Correction: insérer

 $tab[pos] = en_cours$

Trier des cartes

Implémentation

Correction : code complet

```
def tri_insertion(tab: list) -> None:
       tri le tableau dans l'ordre croissant
       for i in range(len(tab)):
 5
          # mémoriser
          en_cours = tab[i]
          pos = i
          # décaler
          while pos > 0 and en_cours < tab[pos-1]:
10
             tab[pos] = tab[pos-1]
11
             pos = pos - 1
          # insérer
13
          tab[pos] = en_cours
14
```

Trier des cartes

Problématique

Trier des cartes manuellement

Fransposer au tri le données

Tri par sélection Implémentation

Correction Complexité

Tri par insertion

Implémentation

ve de correction

Correction : tester

Correction: tester

```
cartes = [i for i in range(1, 14)]
```

shuffle(cartes)
tri_insertion(cartes)

Trier des cartes

Implémentation

67 / 76

Il faut se focaliser sur la boucle interne, non bornée.

Activité 4 : Déterminer un variant de la boucle, qui prouve la terminaison.

Preuve de terminaison

Preuve de terminaison

Il faut se focaliser sur la boucle interne, non bornée.

Activité 4 : Déterminer un variant de la boucle, qui prouve la terminaison.

Trier des cartes

Problématique

Trier des cartes manuellement

nsposer au tri données

mplémentation Ferminaison

Correction

i par insertion

plémentation

Preuve de terminaison

euve de correction mplexité

1 | white pos > 0 and en_cours < tab[pos-1] : tab[pos] = tab[pos-1] : pos = pos-1

pos est un variant de la boucle

Correction

La boucle externe est bornée donc se termine.

```
while pos > 0 and en_cours < tab[pos-1]:
  tab[pos] = tab[pos-1]
   pos = pos - 1
```

pos est un variant de la boucle.

Trier des cartes

Preuve de terminaison

Preuve de correction

Comme pour le tri sélection, avant chaque itération de la boucle externe, la partie gauche du tableau est triée.

3	6	8	4	5
---	---	---	---	---

Code 11 – Insertion de l'élément 4

Trier des cartes

Problématique

Trier des cartes manuellement

ansposer au tri données

par sélection plémentation

Correction Complexité

Tri par insertion
Implémentation

reuve de terminaison

Preuve de terminaison

Preuve de correction

ve de correction plexité

Ļ

3 6 8 5

3 6 8 5

3 4 6 8 5

Code 12 – Insertion de l'élément 4

Trier des cartes

Problématique

rier des cartes anuellement

ansposer au tri données

ri par sélection
Implémentation
Terminaison

Correction Complexité

Implémentation

reuve de terminaison

Preuve de correction

La boucle externe effectue n itérations dans tous les cas.

1 for i in range(len(tab)):

Cependant, le nombre d'itérations de la boucle interne peut varier.

while pos > 0 and en_cours < tab[pos-1]:

3 6 8

Code 13 – Insertion de l'élément 4

roblématique

ier des cartes

Trier des cartes

ansposer au tri données

Implémentation Terminaison Correction

i par insertion mplémentation

euve de terminaison

Preuve de correction Complexité

Activité 5 :

- 1. Compter le nombre d'itérations de la boucle interne si le tableau est déjà trié.
- 2. Compter le nombre d'itérations de la boucle interne si le tableau est trié dans l'ordre décroissant.

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au tri e données

ri par selection Implémentation

orrection omplovité

omplexité i par insertion

Implémentation

euve de terminaison

uve de correction

Correction

while pos > 0 and en_cours < tab[pos-1]:

1	4	5	7	8
---	---	---	---	---

Code 14 – Le tableau est déjà trié. La boucle interne n'effectue aucune itération.

Trier des cartes

Correction

```
for i in range(len(tab)):
    en_cours = tab[i]
    pos = i
    while pos > 0 and en_cours < tab[pos-1]:</pre>
```

```
8 7 5 4 1
```

Code 15 – Le tableau est inversé. La boucle interne effectue i itérations.

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

i par sélection

Correction

Complexité ri par insertion

mplémentation

Preuve de terminaison

Preuve de correction

Complexité

En pratique, tri un peu meilleur que le tri par sélection.

À retenir

Le tri par insertion effectue un nombre moyen d'opérations qui dépend de n^2 .

Problématique

Trier des cartes manuellement

> données par sélection

erminaison

omplexité

plémentation

euve de terminaison