Christophe Viroulaud

Première NSI

poser au tri

Trier des cartes

sélection sentation aison

tion exité

exite insertion

de terminaison

de correction xité

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

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

Implémentation Terminaison

Correction

Complexité

ri par insertion

Implémentation

Preuve de terminaiso

Preuve de terminaisor

reuve de correctio omplexité



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

Activité 1 :

- Prendre le paquet de cartes mélangées et les étaler sur la table
- 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

ransposer au tri e données

ri par sélection Implémentation

orrection

Complexité ri par insertion

plémentation

ve de terminaison ve de correction

Différentes méthodes

Différentes méthodes

► Tri par sélection en place

Differences methodes

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

Trier des cartes

'roblèmatique

Trier des cartes manuellement

ansposer au tr

plémentation

rminaison rrection

mplexité

r insertion

lémentation

uve de terminaison

ive de correction

lexite

Tri par sélection en place Pour chaque carte du tas Trouver la plus petite carte dans la partie Echanger cette carte avec la première de la Code 1 - Tri par sélection (en place)

Tri par sélection en place

Trier des cartes manuellement

Trier des cartes

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 1 – Tri par sélection (en place)

Retour menu

6 / 70

Trier des cartes ☐ Trier des cartes manuellement

Tri par sélection en place



Tri par sélection en place



FIGURE – Sélectionne la plus petite du tas non trié

Retour menu

Trier des cartes

Trier des cartes manuellement

Trier des cartes ☐ Trier des cartes manuellement

Tri par sélection en place



Tri par sélection en place

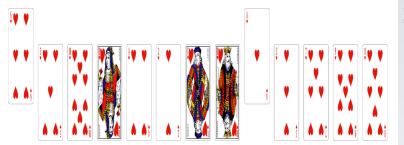


FIGURE – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Trier des cartes manuellement

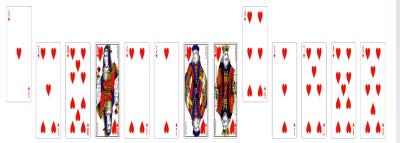
Trier des cartes

Trier des cartes manuellement

└─Tri par sélection en place

Tri par selection en place

Tri par sélection en place



 ${
m Figure}$ – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

e données Fri par sélection

Terminaison Correction

Correction Complexité

Tri par insertion

Implémentation

reuve de terminaison

reuve de correctio omplexité Tri par sélection en place

Tri par sélection en place

Tri par sélection en place

FIGURE – La carte est à sa place

Trier des cartes

Trier des cartes manuellement

Retour menu

└─Tri par sélection en place



Tri par sélection en place

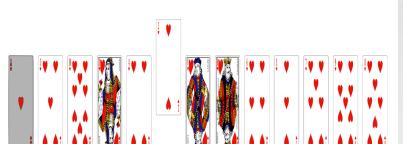


FIGURE – Sélectionne la plus petite du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

de données

Implémentation
Terminaison

Correction

Complexite Tri par insertion

Tri par insertion

Implémentation

reuve de terminaison reuve de correction

omplexité

Trier des cartes

Trier des cartes manuellement

☐Tri par sélection en place



Tri par sélection en place

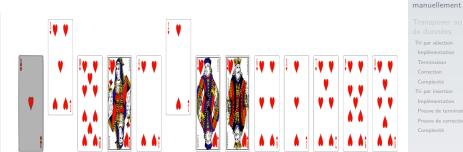


FIGURE – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

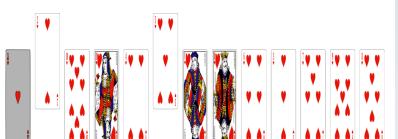
12 / 70

Trier des cartes

└─Tri par sélection en place



Tri par sélection en place



 $\label{eq:Figure} \operatorname{Figure} - \operatorname{\acute{E}change} \ \operatorname{avec} \ \operatorname{la} \ \operatorname{première} \ \operatorname{carte} \ \operatorname{du} \ \operatorname{tas} \ \operatorname{non} \ \operatorname{tri\acute{e}}$

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

e données

Implémentation Terminaison

Complexité

Tri par insertion

Implémentation

Preuve de terminaison

omplexité

Tri par sélection en place

Tri par sélection en place

Tri par sélection en place

Trier des cartes

Trier des cartes manuellement

FIGURE – La carte est à sa place

Retour menu

Pour chaque carte du tas

Trouver la plus petite carte du tableau non trié.

La placer à la fin du tableau tri é.

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 - Trouve la plus petite du tas non trié

Retour menu

Tri par sélection - nouveau tableau

Retour menu

Tri par sélection - nouveau tableau

FIGURE – La place à la fin du tableau trié

Retour menu

Trier des cartes

Trier des cartes manuellement

└─Tri par sélection - nouveau tableau



Tri par sélection - nouveau tableau

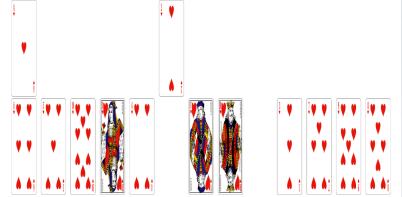


FIGURE – Trouve la plus petite du tas non trié

Retour menu

Trier des cartes

oblématique

Trier des cartes manuellement

Fransposer au tr

Tri par sélection Implémentation Terminaison Correction

Complexité
Tri par insertion

Implémentation
Preuve de terminaise

Preuve de correction

18 / 70

Retour menu

Tri par sélection - nouveau tableau

FIGURE – La place à la fin du tableau trié

Retour menu

Tri par insertion en place

Pour chaque carte du tas

Mémoriser la carte en cours

Décaler vers la droite toutes les cartes précé dentes, supérieures à la carte en cours.

Insérer la carte en cours dans l'espace vide.

Code 3 – Tri par insertion (en place)

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au tri

i par sélection

rminaison

omplexité

par insertion

lémentation uve de terminaison

ve de terminaison ve de correction

lexité

☐Tri par insertion en place

Tri par insertion en place

Tri par insertion en place



FIGURE - Mémoriser la première carte dans le tas non trié

Retour menu

Trier des cartes

manuellement

Trier des cartes

☐Tri par insertion en place

Tri par insertion en place

lci pas de carte triée encore

Tri par insertion en place

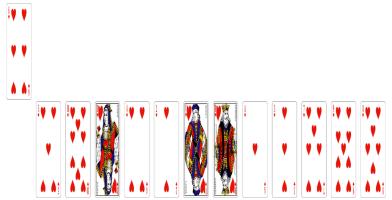


FIGURE – Décaler les cartes supérieures déjà triées

Retour menu

Trier des cartes

Trier des cartes manuellement

la carte est dans le tas trié

☐Tri par insertion en place

Tri par insertion en place

Tri par insertion en place

Trier des cartes

Trier des cartes

FIGURE – Replacer la carte dans l'espace

manuellement

Retour menu

23 / 70















Trier des cartes

Trier des cartes manuellement

└─Tri par insertion en place

la carte est dans le tas trié



Tri par insertion en place

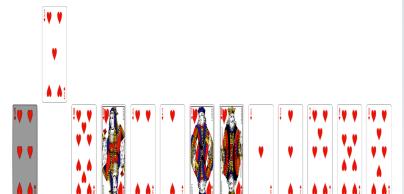


FIGURE – Mémoriser la première carte dans le tas non trié

Retour menu

Trier des cartes

oblématique

Trier des cartes manuellement

ransposer au t

Implémentation Terminaison Correction

Complexité Tri par insertion

Tri par insertion

Implémentation

Preuve de terminaise

Preuve de terminal Preuve de correction

Tri par insertion en place

☐Tri par insertion en place

lci pas de carte triée encore

Tri par insertion en place

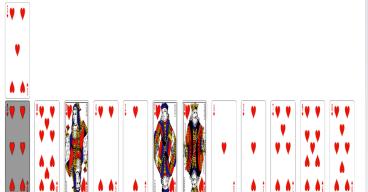


FIGURE – Décaler les cartes supérieures déjà triées

Retour menu

Trier des cartes

Trier des cartes manuellement

la carte est dans le tas trié

☐Tri par insertion en place

Tri par insertion en place

Tri par insertion en place

FIGURE – Replacer la carte dans l'espace

Trier des cartes

manuellement

Trier des cartes

Retour menu

Tri par insertion en place

Retour menu

Tri par insertion en place

olématique

Trier des cartes

Trier des cartes

manuellement

Tri par sélection
Implémentation

Terminaison
Correction
Complexité

Complexité Tri par insertio

Tri par insertio

Implémentation

Preuve de ter

Preuve de correc

FIGURE – Après plusieurs itérations

A;

A: A A:

Retour menu

27 / 70

Retour menu

☐Tri par insertion en place

la carte est dans le tas trié

Tri par insertion en place

FIGURE - Mémoriser la première carte dans le tas non trié

Retour menu

Trier des cartes manuellement

Trier des cartes

Tri par insertion en place Retour menu

☐Tri par insertion en place

lci pas de carte triée encore

Tri par insertion en place

FIGURE – Décaler les cartes supérieures déjà triées

Retour menu

Trier des cartes

Trier des cartes manuellement

Retour menu

Tri par insertion en place

Tri par insertion en place

Trier des cartes

Trier des cartes

manuellement

FIGURE – Insérer la carte dans l'espace

Retour menu

Pour chaque carte du tas

Prendre la première carte du tableau non tri é.

Dans le tableau tri é, 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

Trier des cartes

Problématique

Trier des cartes manuellement

Fransposer au tri

par sélection

orrection

par insertion

nplémentation

ıve de terminaison

ive de correction

Tri par insertion - nouveau tableau

Trier des cartes

Trier des cartes

manuellement

32 / 70

Tri par insertion - nouveau tableau

Trier des cartes

Trier des cartes

manuellement

FIGURE – Décaler les cartes supérieures du tableau trié

Retour menu



Tri par insertion - nouveau tableau

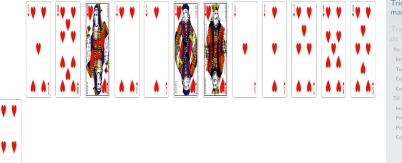


FIGURE – Insérer la carte dans le tableau triée

Retour menu

Trier des cartes

Trier des cartes

manuellement

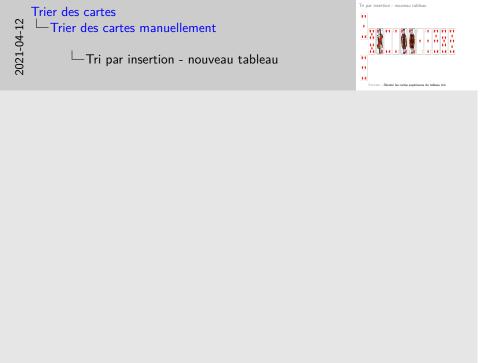




Trier des cartes

FIGURE – Prendre la première carte non triée

35 / 70



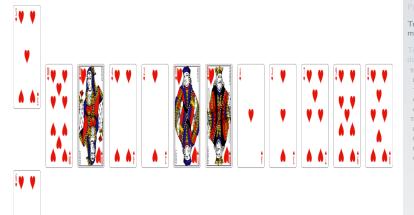


FIGURE – Décaler les cartes supérieures du tableau trié

Trier des cartes

Trier des cartes

manuellement

36 / 70

└─Tri par insertion - nouveau tableau



Tri par insertion - nouveau tableau

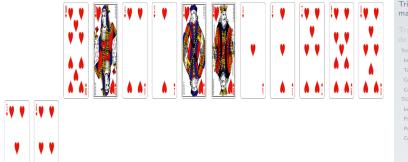


FIGURE – Insérer la carte dans le tableau triée

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au tri

Tri par sélection Implémentation

Terminaison Correction

Complexité
Tri par incortion

Tri par insertion

Implémentation

Preuve de terminaiso

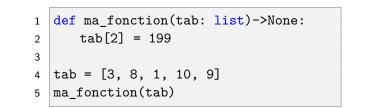
uve de correction nplexité

Trier des cartes —Transposer au tri de données



- 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.



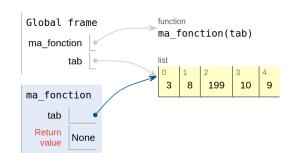


FIGURE – La fonction modifie le tableau original

Problématique

Trier des cartes

Trier des cartes

Transposer au tri

de données
Tri par sélection
Implémentation
Terminaison

orrection omplexité i par insertion nolémentation

reuve de correction omplexité Activité 2 :

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

Tri par sélection - question préparatoire

Tri par sélection - question préparatoire

Activité 2 :

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

Trier des cartes

Problématique

Trier des cartes manuellement

Fransposer au tri le données

par sélection

Implémentation

orrection

omplexité

par insertion

uve de terminaison

uve de correction nplexité

```
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

Trier des cartes

ransposer au tri e données

par sélection

Implémentation

Terminaison

Complexité

par insertion

nplémentation

reuve de terminaison reuve 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

```
def trouver_mini(tab: list , i_depart: int) -> int:
    """
    renvoie | '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

roblématique

Trier des cartes nanuellement

ransposer au tri e données

i par sélection

Implémentation

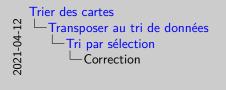
rminaison prrection

omplexité

lémentation

uve de terminaison

ive de correction iplexité



```
def echanger(tab: list . i: int . i: int) -> None:
   tab[i], tab[j] = tab[j], tab[i]
```

Trier des cartes

Implémentation

```
inverse les élément d'indices i et j du tableau
3
      tab[i], tab[j] = tab[j], tab[i]
```

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

Trier des cartes

Problèmatique

Trier des cartes nanuellement

ransposer au tri e données

i par sélection

Implémentation

Terminaison

omplexité

par insertion

nplémentation

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*.

Droblómatique

Trier des cartes

Trier des cartes manuellement

ransposer au tri e données

Tri par sélection

erminaison

orrection omplexité

nplémentation

euve de terminaison

e de correction lexité

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

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

Trier des cartes

Implémentation

Preuve de terminaison : variant de boucle

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

de de deux boucles bornées donc qui terminent.

de de deux composé Tri par insert Implémenta Prave de tri Preuve de tri Preuve de Complexité

atique

des cartes

Trier des cartes

ansposer au tri

sélection mentation

rection nplexité

par insertion

plémentation euve de terminaison

uve de terminaison uve de correction

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

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

Code 5 – Avant la première itération, la partie gauche est vide, donc triée.

Trier des cartes

Correction

Preuve de correction : invariant de boucle

8 3

Code 6 – Avant la deuxième itération, la partie gauche est triée.

Preuve de correction : invariant de boucle

Trier des cartes

Correction

Preuve de correction : invariant de boucle

Preuve de correction : invariant de boucle

8

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

Trier des cartes

Correction

Efficacité du tri

1 for j in range(i_depart+1, len(tab)):
2 if tab[j] < tab[i_mini]:
3 i_mini = j</pre>

Efficacité du tri

- ▶ 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.
- **...**

```
for j in range(i_depart+1, len(tab)):
    if tab[j] < tab[i_mini]:
        i_mini = j</pre>
```

Trier des cartes

Problématique

Trier des cartes nanuellement

Transposer au tri le données

ri par sélection Implémentation

orrection omplovité

Complexité

nplémentation

Preuve de terminaison

uve de correction nplexité $\sum_{n=1}^{n-1} k = 1 + 2 + \cdots + n - 1 = \frac{n \cdot (n-1)}{2}$

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

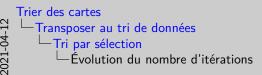
$$\sum_{k=1}^{n-1} k = 1 + 2 + \dots + n - 1 = \frac{n \cdot (n-1)}{2}$$

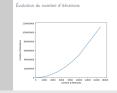
À retenir

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

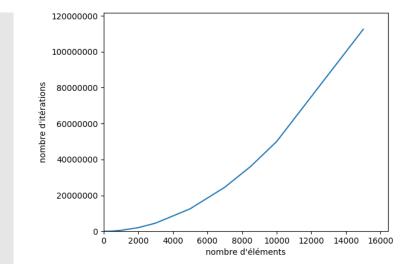
Complexité





15000 éléments \rightarrow 100 millions d'itérations

Évolution du nombre d'itérations



Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au tri données

> ar sélection lémentation

Correction
Complexité

par insertion

plémentation

ve de terminaison
ve de correction

Activité 3 : L Écrire la fonction tri_insertion(tab : list) →None en s'appuyant sur l'algorithme. Les indications suivantes nermettront de construire les en cours de placement et pos, position actuelle de décaler les éléments vers la droite.

Tri par insertion

Tri par insertion

Activité 3:

- 1. 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.
 - ► Insérer : placer l'élément *en_cours* à la nouvelle position pos.
- 2. Tester la fonction sur un tableau.

Trier des cartes

Implémentation

54 / 70

Rappel de l'algorithme

Rappel de l'algorithme

4

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

Code 8 – Tri par insertion (en place)

Trier des cartes

Implémentation

Correction : boucle principale

Correction : boucle principale

```
def tri_insertion (tab: list ) -> None:

tri le tableau dans l'ordre croissant

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

Trier des cartes

Problématique

Frier des cartes nanuellement

Fransposer au tri le données

par sélection

Terminaison Correction

Complexité
Tri par insertion

Implémentation

reuve de terminaison

ve de correction

 $en_cours = tab[i]$

pos = i

Correction: mémoriser

en_cours = tab[i]

Trier des cartes

Implémentation

57 / 70

Correction : décaler

Correction : décaler

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

Problèmatique

Trier des cartes

Trier des cartes

ransposer au tri e données

IONNEES
ar sélection
lémentation

rminaison

omplexité par insertion

Tri par insertion

Implémentation

lémentation uve de terminaison

e de terminaison e de correction 2021-04-12

Correction: insérer

tab[pos] = en_cours

Trier des cartes

Implémentation

```
Correction: code complet

[inf str_mercine (tab. init) => None

to this behave dear indice consume

for in recognitive(bab)

for in recognitive(bab)

for in recognitive(bab)

par = [
```

Correction : code complet

```
def tri_insertion (tab: list ) -> None:
        tri le tableau dans l'ordre croissant
        for i in range(len(tab)):
            # 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

ri par sélection Implémentation

Correction Complexité

Tri par insertion

nplementation reuve de terminaison

uve de terminaison uve de correction Correction : tester

Correction: tester

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

Trier des cartes

Implémentation

prouve la 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

rier des cartes anuellement

nsposer au tri données

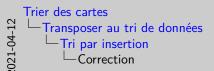
prementation rminaison rrection

mplexité par insertion

plémentation

Preuve de terminaison

ve de terminaison



| while par > 0 and m_nows < talpm=1|
| talpm= = talpm=-1|
| max par-1|
| par ext on varient de la foscife.

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</pre>
```

pos est un variant de la boucle.

Trier des cartes

'roblèmatique

Trier des cartes nanuellement

ransposer au tri e données

par sélection plémentation

rection

ri par insertion

Implémentation

Preuve de terminaison

lexité



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



Code 9 – Insertion de l'élément 4

Trier des cartes

Preuve de correction

6

Code 10 – Insertion de l'élément 4

Trier des cartes

Preuve de correction

La boucle externe effectue n itérations dans tous les cas.

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]:

6 8 5

Code 11 – Insertion de l'élément 4

roblématique

Trier des cartes

Trier des cartes

ansposer au tri données

pplémentation erminaison orrection

i par insertion

mplémentation

reuve de terminaison

Preuve de correction Complexité

Lomplexite

Activité 5 :

ctivité 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

Activité 5 :

- 1. Compter le nombre d'itérations de la boucle interne si le tableau est déjà trié.
- 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

Implémentation

orrection

i par insertion

mplémentation

euve de terminaison

Complexité

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

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

Code 12 – Le tableau est déjà trié. La boucle interne n'effectue aucune itération.

Trier des cartes

Problématique

Trier des cartes

ransposer au tri e données

par sélection plémentation

Correction Complexité

ri par insertion Implémentation

Preuve de terminaison

Complexité

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

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

```
5
      4
```

Code 13 – Le tableau est inversé. La boucle interne effectue i itérations.

Trier des cartes

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 .

Trier des cartes

oblématique

rier des cartes anuellement

> nsposer au tri données

plémentation rminaison

rrection mplevité

ar insertion

lémentation uve de terminaison

ive de correction

Complexité