Première - NSI

Trier des cartes

Christophe Viroulaud

Première - NSI

blematique

Trier des cartes

anuellement

onnées r sélection

émentation ninaison

ection plexité

nplexité ar insertion

émentation

e de terminaisor e de correction

Sommaire

- 1. Problématique

Implémentation

Terminaison

Correction

Complexité Implémentation

Preuve de terminaison

Preuve de correction Complexité

Problématique

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

FIGURE 1 – Cartes mélangées

Trier des cartes Trier des cartes manuellement

Activité 1 :

- Prendre le paquet de cartes mélangées et les étaler
 cor le table
- 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

Fransposer au tri le données

par sélection

orrection

omplexité

plémentation

ve de terminaison ve de correction

Différentes méthodes

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

Différentes méthodes

► Tri par sélection en place

Trier des cartes

Trier des cartes manuellement

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

Transposer au 1

e données ri par sélection

Terminaison

Complexité

Implémentation

reuve de terminaison



Tri par sélection en place

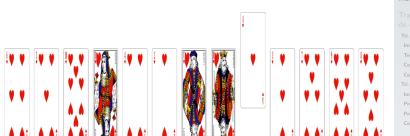


FIGURE 2 – Sélectionne la plus petite du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

Fri par sélection

Terminaison Correction

Complexité

Tri par insertion

Implémentation

reuve de terminaison

Tri par sélection en place

**Tripar se l'acceptant de preside sere de les une tel

**Rétair mais

Tri par sélection en place

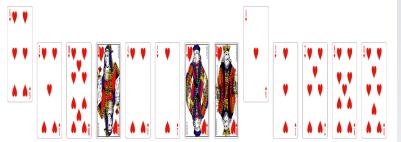


FIGURE 3 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Fransposer au tr

e donnees ri par sélection Implémentation Terminaison

Correction
Complexité
Tri par insertion

Implémentation

Preuve de terminaison

Tri par effection en place

Forms 4 - Calego are la puede orte de la remotid

Réformess

Tri par sélection en place

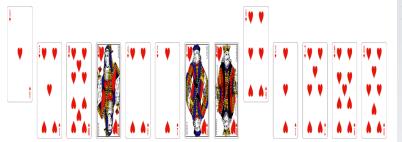


FIGURE 4 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

e données

Tri par sélection

Implémentation

Terminaison
Correction
Complexité

Tri par insertion

Implémentation

reuve de terminaison

Retour menu

FIGURE 5 – La carte est à sa place

Tri par sélection en place

Retour menu



Tri par sélection en place

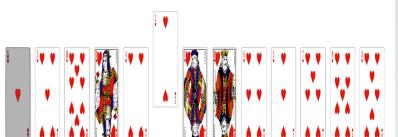


FIGURE 6 – Sélectionne la plus petite du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

données i par sélection

Terminaison Correction

Complexité Tri par insertion

Iri par insertion
Implémentation

Preuve de terminaisor

euve de correction

Tri par sélection en place

FULTIME 7 - Change avec la premiere carte de tan non tole

Relianar manus

Tri par sélection en place

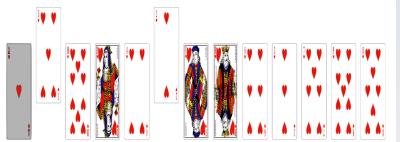


FIGURE 7 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Problématique

Trier des cartes manuellement

Transposer au tri

ri par sélection
Implémentation

Terminaison Correction

Complexité

Tri par insertion

Preuve de terminaisor

Preuve de terminaison



Tri par sélection en place

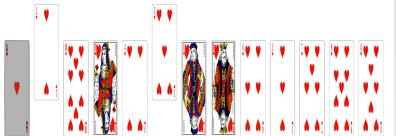


FIGURE 8 – Échange avec la première carte du tas non trié

Retour menu

Trier des cartes

Trier des cartes manuellement

Retour menu

FIGURE 9 – La carte est à sa place

Tri par sélection en place

Retour menu

manuellement

Tri par sélection - nouveau tableau

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

Problématique

Trier des cartes

Trier des cartes manuellement

Transposor ou tri

ransposer au tri e données

ri par sélection Implémentation

Terminaison Correction

Complexité

ar insertion

plementation euve de terminaison

uve de terminaison uve de correction

olexité

└─Tri par sélection - nouveau tableau



Tri par sélection - nouveau tableau

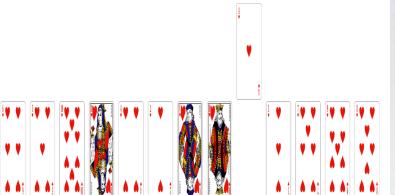


FIGURE 10 – Trouve la plus petite du tas non trié

Retour menu

Trier des cartes

oblématique

Trier des cartes

manuellement

le données
Tri par sélection

Correction Complexité Tri par insertion

Implémentation

Preuve de termina

Preuve de terminais

Preuve de correctior Complexité

Tri par sélection - nouveau tableau

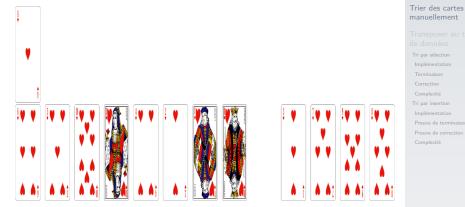
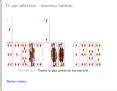


FIGURE 11 – La place à la fin du tableau trié

Retour menu



Tri par sélection - nouveau tableau

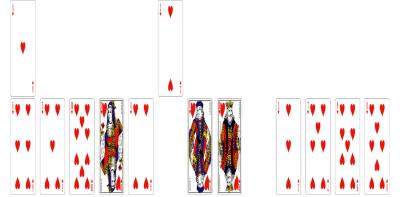


FIGURE 12 – Trouve la plus petite du tas non trié

Retour menu

Trier des cartes

roblématique

Trier des cartes

manuellement

Fransposer au tr

Tri par sélection
Implémentation
Terminaison
Correction

Tri par insertion

Preuve de terminaisor
Preuve de correction

└─Tri par sélection - nouveau tableau



Tri par sélection - nouveau tableau

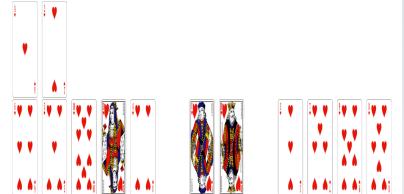


FIGURE 13 – La place à la fin du tableau trié

Retour menu

Trier des cartes

oblématique

Trier des cartes

manuellement

de données
Tri par sélection
Implémentation

Correction
Complexité
Tri par insertion

Tri par insertion Implémentation

Preuve de terminais

reuve de correction Complexité

Tri par insertion en place

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

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au t

ri par sélection

orrection

ar insertion

uve de terminaison

reuve de correction

nplexité

☐Tri par insertion en place

2021-05-25

Financia 1 - Memerier la previen carie dans la se ana sisi

Tri par insertion en place

Tri par insertion en place

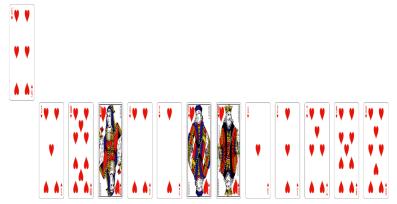


FIGURE 14 – Mémoriser la première carte dans le tas non trié

Retour menu

ématique

Trier des cartes

Trier des cartes

manuellement

ansposer au tri

mplémentation Ferminaison Correction

Complexité Tri par insertion

Tri par insertion

Implémentation
Preuve de terminai

Preuve de termina Preuve de correcti

Tri par insertion en place

Tri par insertion en place

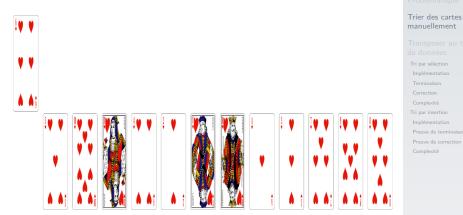


FIGURE 15 – Décaler les cartes supérieures déjà triées

Retour menu

.

la carte est dans le tas trié

☐Tri par insertion en place

2021-05-25

Tri par insertion en place

Tri par insertion en place

Trier des cartes

Trier des cartes manuellement

FIGURE 16 – Replacer la carte dans l'espace

Retour menu

└─Tri par insertion en place

la carte est dans le tas trié



Tri par insertion en place



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

Retour menu

Trier des cartes

oblématique

Trier des cartes manuellement

nanuellement

ansposer au tri données i par sélection

nplémentation erminaison orrection

Complexité Tri par insertion

Tri par insertion
Implémentation

Implémentation
Preuve de terminai

Preuve de correcti

omplexité

Tri par insertion en place

☐Tri par insertion en place

lci pas de carte triée encore

Tri par insertion en place

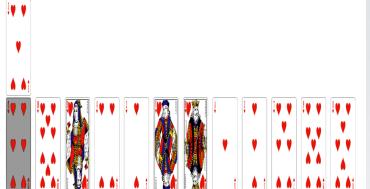


FIGURE 18 – Décaler les cartes supérieures déjà triées

Retour menu

Trier des cartes

Trier des cartes manuellement

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 manuellement

FIGURE 19 – Replacer la carte dans l'espace

Retour menu



Tri par insertion en place

Tri par insertion en place

Trier des cartes

Trier des cartes manuellement

FIGURE 20 – Après plusieurs itérations

Retour menu



Tri par insertion en place

Trier des cartes

Trier des cartes manuellement

Retour menu



Tri par insertion en place

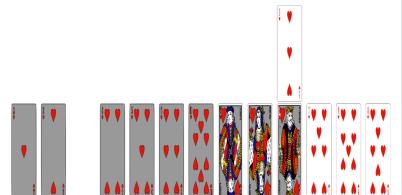


FIGURE 22 – Décaler les cartes supérieures déjà triées

Retour menu

Trier des cartes

manuellement

Florina 22 - healers is contributed from Fragues

Tri par insertion en place

Tri par insertion en place

Trier des cartes manuellement

Transposer au t de données

Iri par sélection
Implémentation
Correction
Complexité
Tri par insertion
Implémentation
Preuve de terminaison
Preuve de correction

FIGURE 23 – Insérer la carte dans l'espace

Retour menu

Tri par insertion - nouveau tableau

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

Fransposer au ti

e données Fri par sélection

Terminaison Correction

Complexité ri par insertion

nplémentation

reuve de terminaison

reuve de correction

Tri par insertion - nouveau tableau

2021-05-25

Tri par insertion - nouveau tableau

Tri par insertion - nouveau tableau

Trier des cartes

Trier des cartes

manuellement

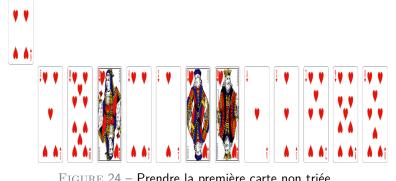


FIGURE 24 – Prendre la première carte non triée

Retour menu

└─Tri par insertion - nouveau tableau



Tri par insertion - nouveau tableau

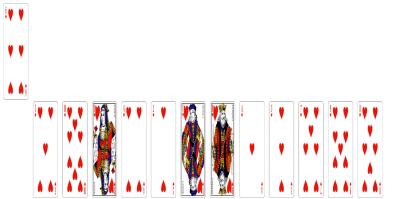


FIGURE 25 – Décaler les cartes supérieures du tableau trié

Retour menu

Trier des cartes

oblématique

Trier des cartes

manuellement

Transposer au tr de données

Terminaison Correction

Complexité Tri par insertion

Iri par insertion Implémentation

Preuve de terminaiso

reuve de correction Complexité

Tri par insertion - nouveau tableau



Tri par insertion - nouveau tableau

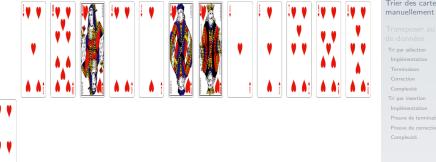


FIGURE 26 – Insérer la carte dans le tableau triée

Retour menu

Trier des cartes



Tri par insertion - nouveau tableau

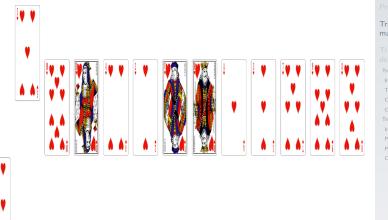
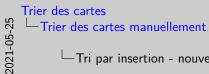


FIGURE 27 – Prendre la première carte non triée

Trier des cartes manuellement

Trier des cartes

37 / 76



Tri par insertion - nouveau tableau



Tri par insertion - nouveau tableau

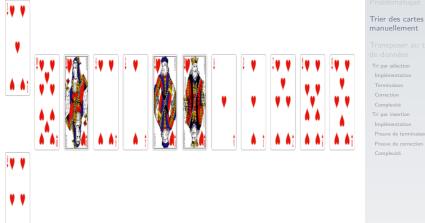


FIGURE 28 – Décaler les cartes supérieures du tableau trié

Trier des cartes

Retour menu

38 / 76

└─Tri par insertion - nouveau tableau



Tri par insertion - nouveau tableau

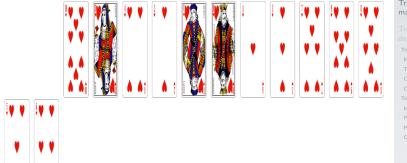


FIGURE 29 – Insérer la carte dans le tableau triée

Retour menu

Trier des cartes

Problématique

Trier des cartes

manuellement

Fransposer au tr le données

Implémentation

Correction

Tri par insertion

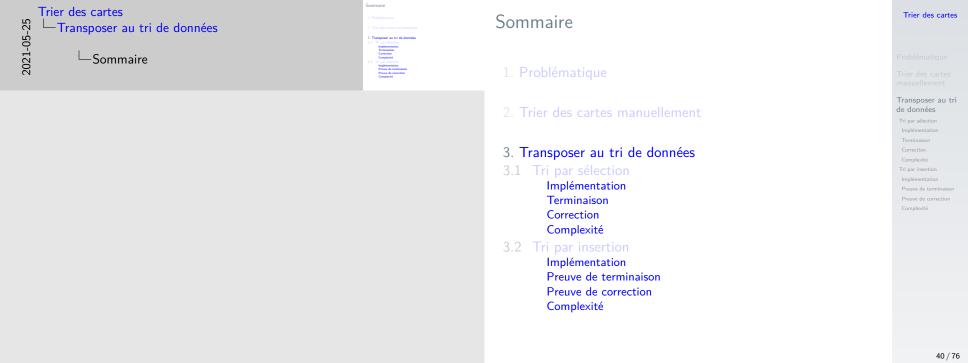
Implémentation

Implémentation

Preuve de terminaiso

Preuve de terminaisc Preuve de correction

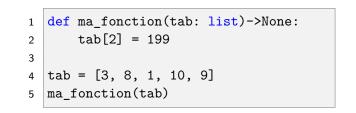
euve de correction mplexité



Trier des cartes Transposer au tri de données

- Quand on passe on tablesce are agreement 2 and fonction control densities required by a tablesce original to tablesce original to tablesce original to table 2 and 2 and
- 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.



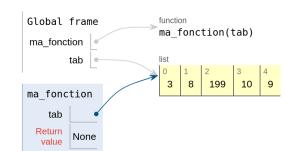


FIGURE 30 – La fonction modifie le tableau original

Problématique

Trier des cartes

Trier des cartes

Transposer au tri

ri par sélection
Implémentation
Terminaison

i par insertion
mplémentation
Preuve de terminaison

41 / 76

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

Problématique

Trier des cartes manuellement

nsposer au tri données

Tri par sélection Implémentation

Terminaison Correction

Fri par insertion

nplémentation reuve de terminaison

uve de terminaison uve de correction

exité

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

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

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

Trier des cartes

ransposer au tri e données

par sélection

Implémentation

Correction

omplexité

nplémentation

uve de terminaison

euve 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

```
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

ransposer au tri e données

ri par sélection

Implémentation

Correction

par insertion

nplémentation

reuve de terminaison

ve de correction plexité



```
def echanger(tab: list, i: int, j: int) -> None:
      inverse les élément d'indices i et j du tableau
3
      tab[i], tab[j] = tab[j], tab[i]
```

Trier des cartes

Trier des cartes

Problèmatique

rier des cartes nanuellement

ransposer au tri e données

i par sélection

Implémentation

erminaison

mnlexité

par insertion

npiementation reuve de terminaison

reuve de terminaison

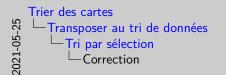
uve de correction nplexité Tri par sélection - Tester

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.

Trier des cartes



```
Correction
```

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

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

```
Trier des cartes
```

Preuve de terminaison : variant de boucle

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

Trier des cartes

Terminaison

Preuve de correction : invariant de boucle

Trier des cartes

Correction

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

8

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

Preuve de correction : invariant de boucle

Preuve de correction : invariant de boucle

8 3

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

Trier des cartes

Correction

Preuve de correction : invariant de boucle

8

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

i par sélection

Correction

Complexité

Tri par insertion

Implémentation

Implémentation

Preuve de terminaiso

euve de terminaison euve de correction 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

Problématique

Trier des cartes manuellement

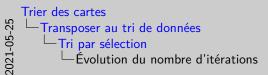
ransposer au tri e données

> par sélection plémentation

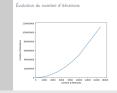
Correction
Complexité

ri par insertion mplémentation

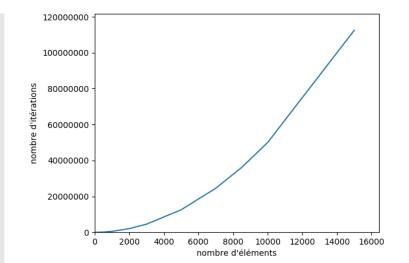
plémentation euve de terminaison



15000 éléments \rightarrow 100 millions d'itérations



Évolution du nombre d'itérations



Trier des cartes

Problématique

Trier des cartes manuellement

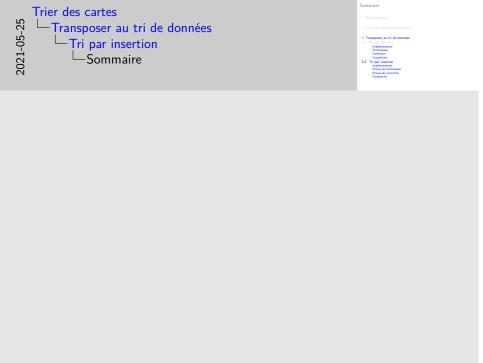
nsposer au tri données

plémentation

Complexité

oar insertion olémentation

e de terminaison e de correction



Sommaire

- 3. Transposer au tri de données
- - Implémentation Terminaison
- Correction
- Complexité
- 3.2 Tri par insertion

Complexité

- Implémentation
- Preuve de terminaison
- Preuve de correction

- Tri par insertion

Trier des cartes

58 / 76

Activité 3:

Eérie la fonction tri_missertion(tab : list)

—Nome en s'appoyant sur l'algorithme. Les
indications suivantes permittente de construire les
trois étapes:

—le fonction de l'appoyant sur visible en_cours, dieman

me care de planement et pas, paulion exterile de
cet different sur visible en_cours, dieman

en care de planement et pas, paulion exterile de
cet different

» Elle de l'appoyant en de l'appoyant de

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

i par sélection nplémentation

Correction

Complexité

Tri par insertion

Implémentation

euve de terminaison euve 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.

Code 9 – Tri par insertion (en place)

Trier des cartes

Problématique

Trier des cartes manuellement

ransposer au tri e données

ri par sélection

Correction

Complexité

ri par insertion

Implémentation

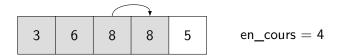
euve de terminaison

plexité



Subtilité sur le tableau Python









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

Trier des cartes

Correction : boucle principale

Correction : boucle principale

Trier des cartes

Problématique

Trier des cartes

ransposer au tri e données

par sélection

Terminaison Correction

Complexité
Tri par incertion

Implémentation

reuve de terminaison

e de correction lexité Correction: mémoriser

en_cours = tab[i] pos = i

Trier des cartes

 $en_cours = tab[i]$ pos = i

Correction : décaler

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

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

pos = pos - 1

Trier des cartes

Correction: insérer

 $tab[pos] = en_cours$

Trier des cartes

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

Transposer au tri de données

> ri par sélection Implémentation

Correction Complexité

Fri par insertion

Implémentation

euve de terminaison euve de correction Correction : tester

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

Trier des cartes

Preuve de terminaison

Il faut se focaliser sur la boucle interne, non bornée

Activité 4 : Déterminer un variant de la boucle, qui

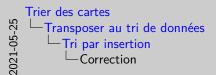
Il faut se focaliser sur la boucle interne, non bornée.

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

Trier des cartes

Preuve de terminaison

68 / 76



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 nanuellement

ransposer au tri e données

ri par sélection mplémentation

orrection

ri par insertion

Implémentation

euve de terminaison

Preuve de correction

3 6 8 5

4

3 6 8 5

3 4 6 8 5

Code 12 – Insertion de l'élément 4

Diematique

Trier des cartes

anuellement

nsposer au tri données par sélection

erminaison orrection

i par insertion

Preuve de terminaisor

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

Problématique

rier des cartes anuellement

Trier des cartes

ansposer au tri données

Implémentation Terminaison Correction

omplexité i par insertion

mplémentation

euve de correction

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 dépositement.

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 sélection Implémentation

Correction

Complexité

mplémentation

euve de terminaison

uve de 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

Problématique

Trier des cartes

ransposer au tri e données

i par sélection

Terminaison Correction

omplexité

Tri par insertion

Preuve de terminaison

Preuve de 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

ransposer au tri

ri par sélection

Terminaison Correction

Complexité

plémentation

euve de terminaison

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 .

roblématique

Trier des cartes manuellement

> données par sélection

rminaison

mplexité

par insertion plémentation

euve de terminaison

Preuve de correction Complexité