Algorithmique avancée

Algorithmes de tris

Frédéric Guyomarch

IUT-A
Université de Lille, Sciences et Technologies

2018/2019 - Semestre 3

Qu'est-ce qu'un tri ?

Definition

Trier, c'est ranger des éléments en respectant un ordre particulier.

Qu'est-ce qu'un tri ?

Definition

Trier, c'est ranger des éléments en respectant un ordre particulier.

Le plus souvent, on trie suivant l'ordre numérique ou lexicographique.

Pourquoi trier ?

Quelques exemples

Le tri est omniprésent en informatique :

- Pour améliorer la recherche d'éléments
  - dichotomie sur tableaux ordonnés
- Pour ranger des éléments en fonction de leurs caractéristiques
  - boutique en ligne : tri par prix ou note
  - emails : tri par taille ou priorité

Comparaison d'algorithmes

Quelques critères

On peut mesurer l'efficacité d'un algorithme en se penchant sur

- le nombre d'étapes nécessaires,
- le nombre de comparaisons réalisées,
- le nombre d'échanges effectués.

Comparaison d'algorithmes

Quelques critères

On peut mesurer l'efficacité d'un algorithme en se penchant sur

- le nombre d'étapes nécessaires,
- le nombre de comparaisons réalisées,
- le nombre d'échanges effectués.

Généralement, on regarde le nombre d'étapes nécessaires dans le pire cas pour comparer deux algorithmes.

Comparaison d'algorithmes

Exemple

Rappelez-vous, pour un tableau ordonné :

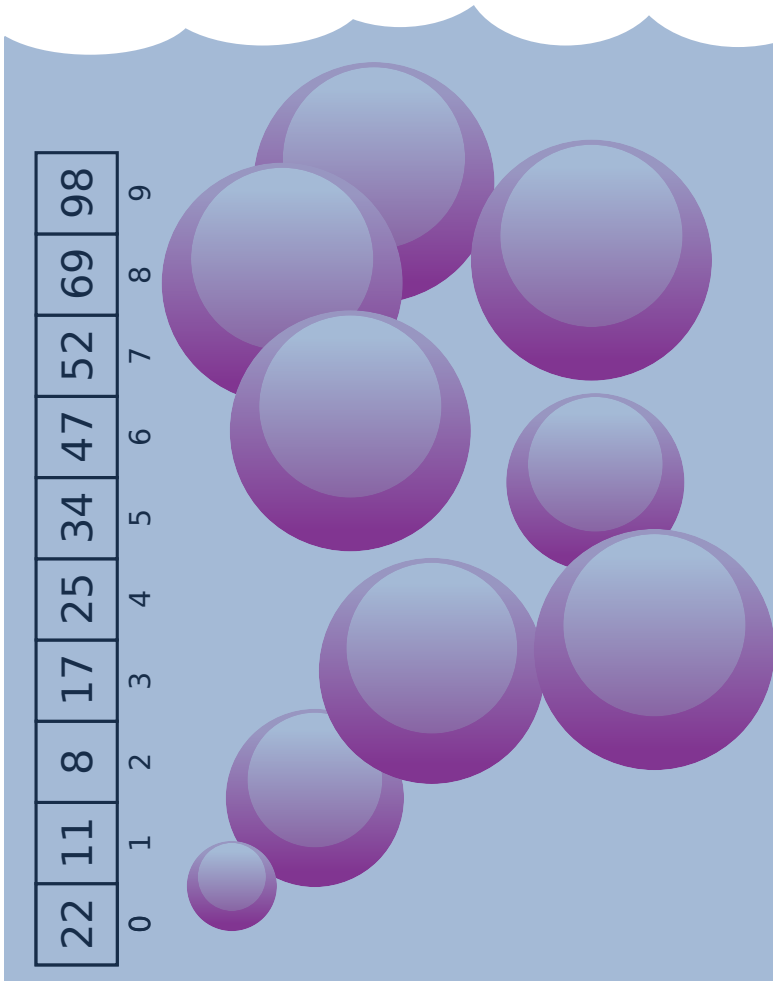
- La recherche séquentielle requiert n étapes dans le pire cas.
- La recherche dichotomique requiert $log_2 n$ étapes dans le pire cas.

Plan

Revue de quelques algorithmes de tri :

- Tri à bulles
- Tri par sélection
- Tri par insertion

Le tri à bulles



Principe

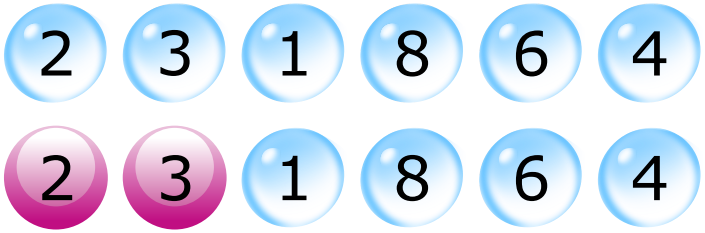
1. On compare successivement les cases deux à deux.
2. On échange les valeurs si nécessaires de manière à respecter l'ordre
3. On répète les étapes 1 et 2 conjointement jusqu'à obtenir un tableau trié.

En images

Un premier tour :

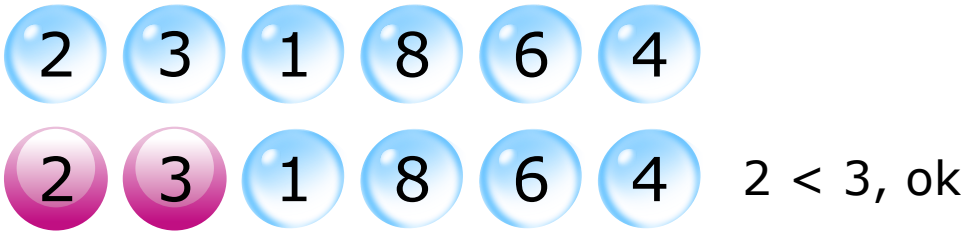
En images

Un premier tour :



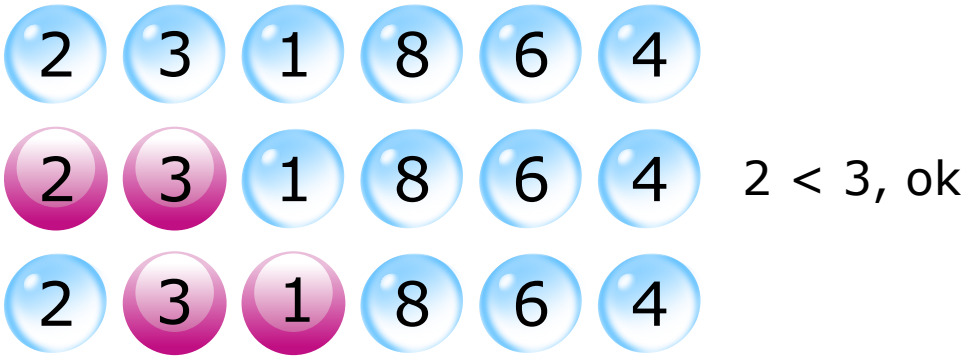
En images

Un premier tour :



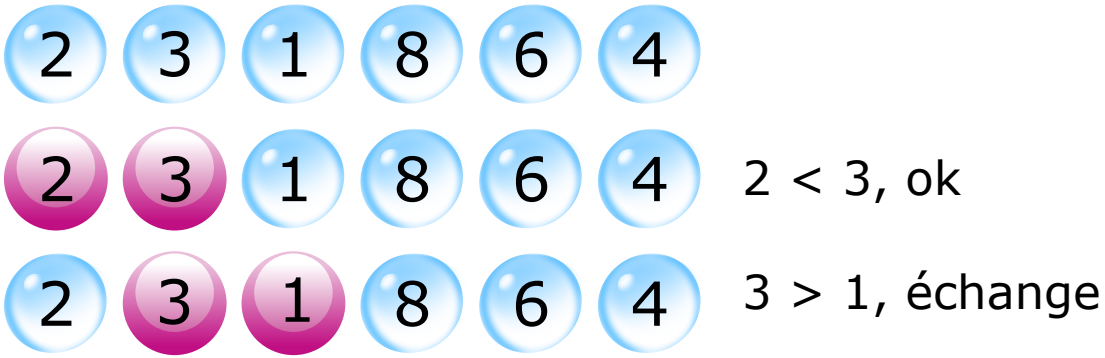
En images

Un premier tour :



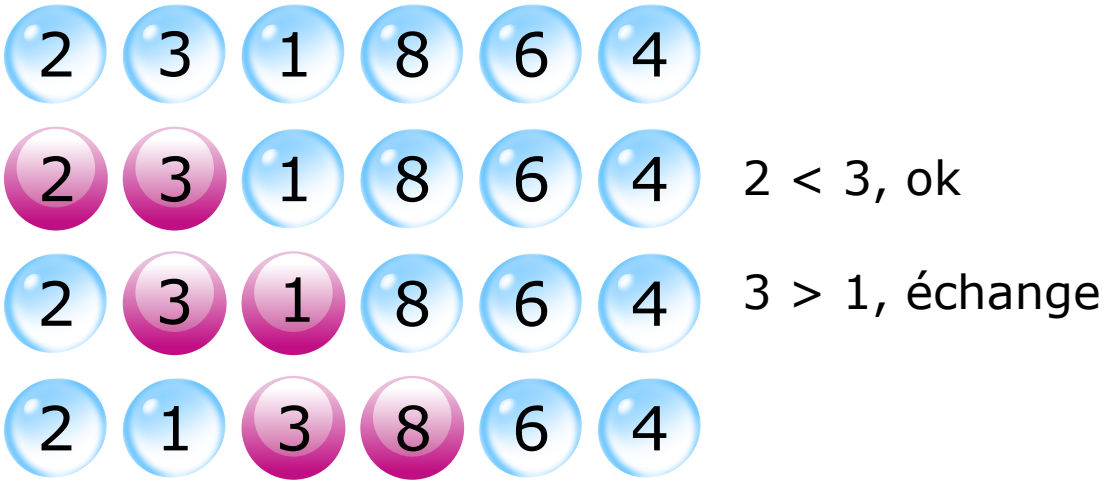
En images

Un premier tour :



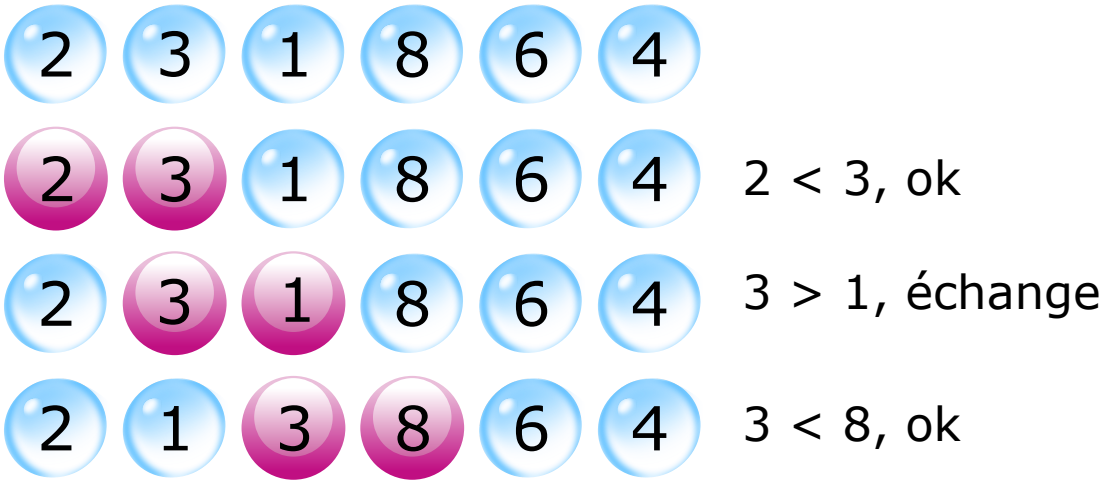
En images

Un premier tour :



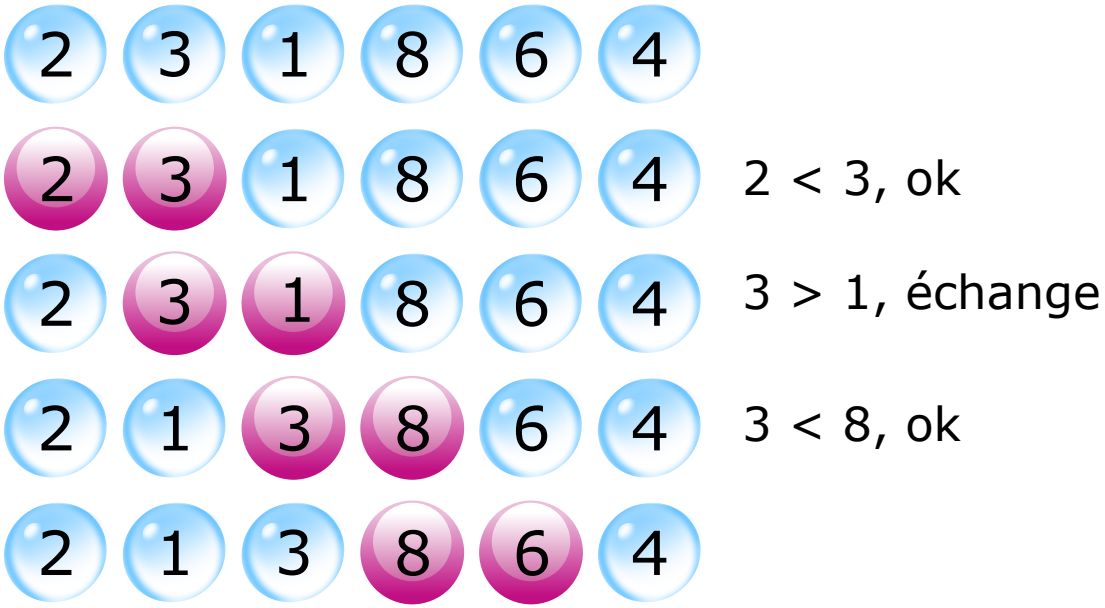
En images

Un premier tour :



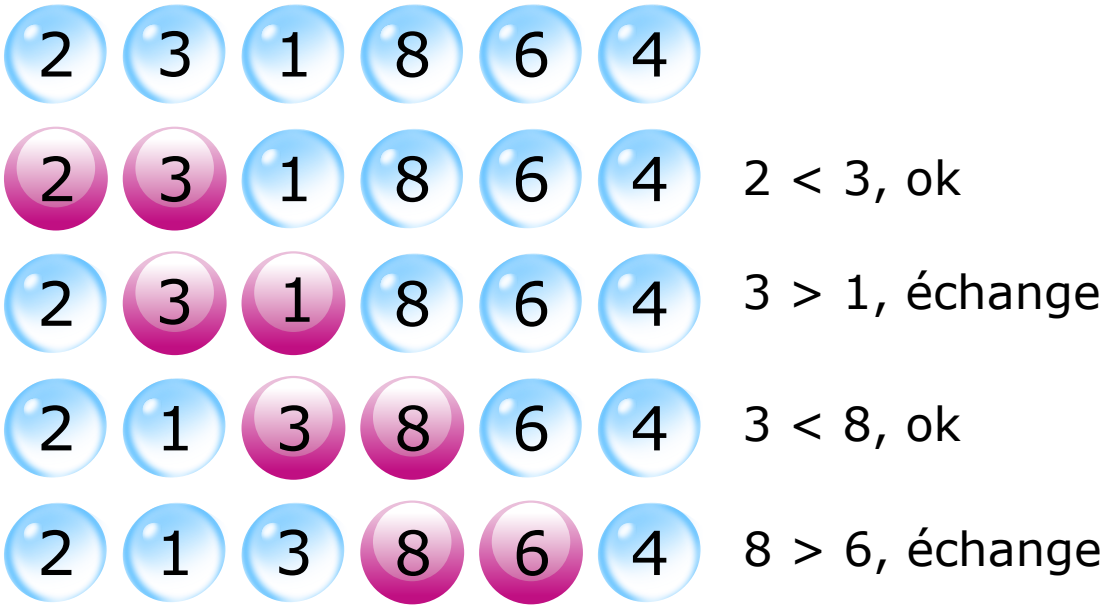
En images

Un premier tour :



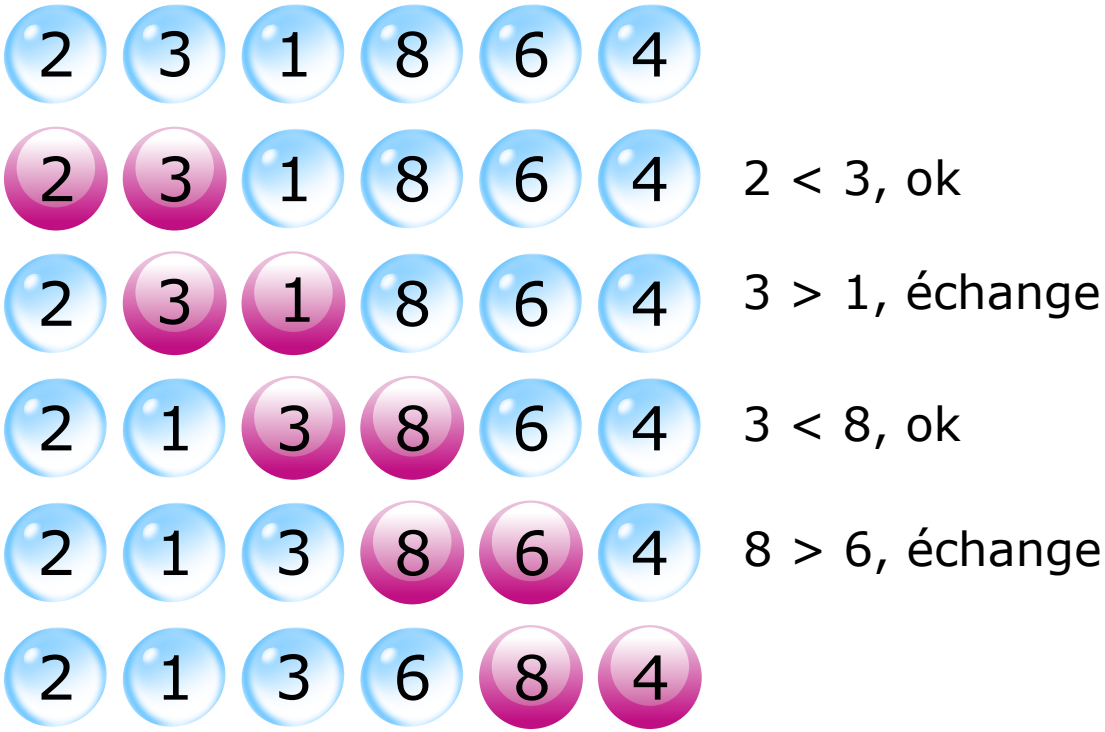
En images

Un premier tour :



En images

Un premier tour :

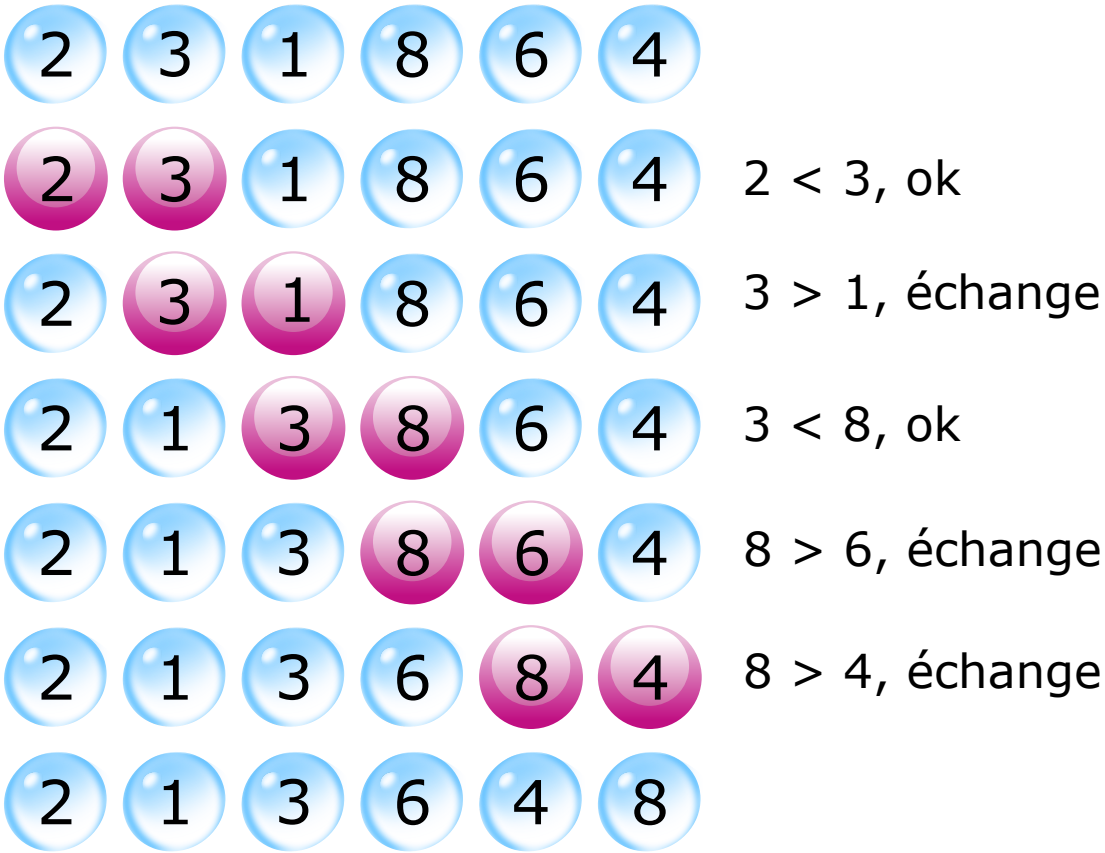


En images

Un premier tour :

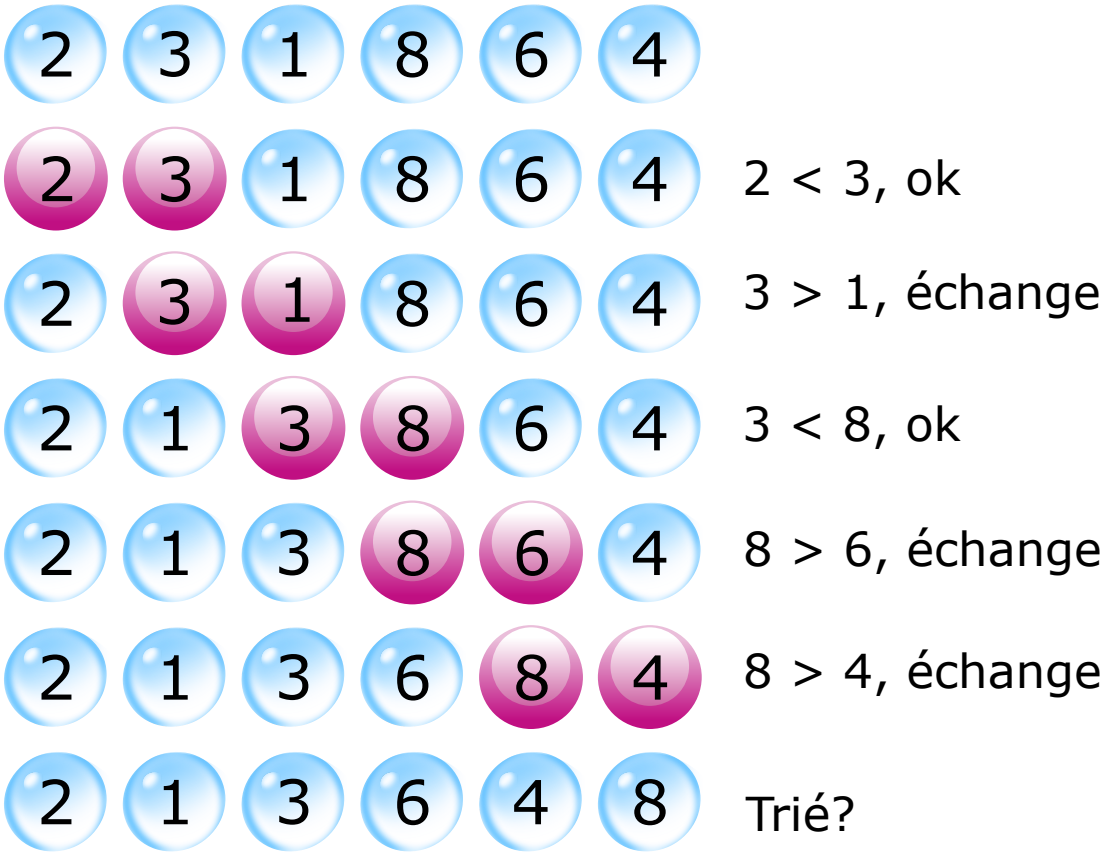
En images

Un premier tour :



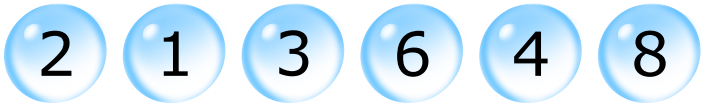
En images

Un premier tour :



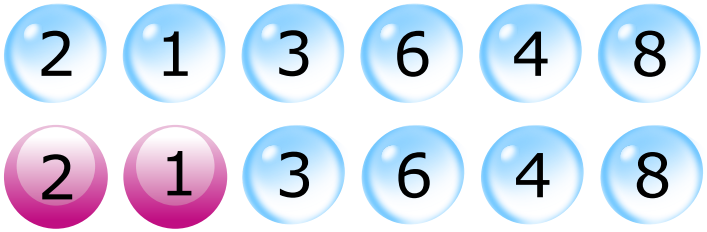
Principe

Un deuxième tour :



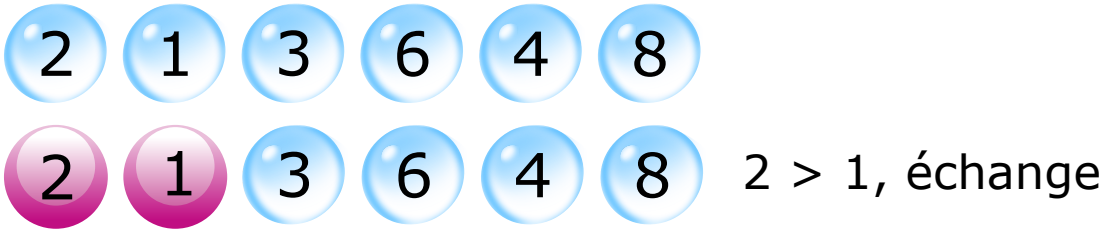
Principe

Un deuxième tour :



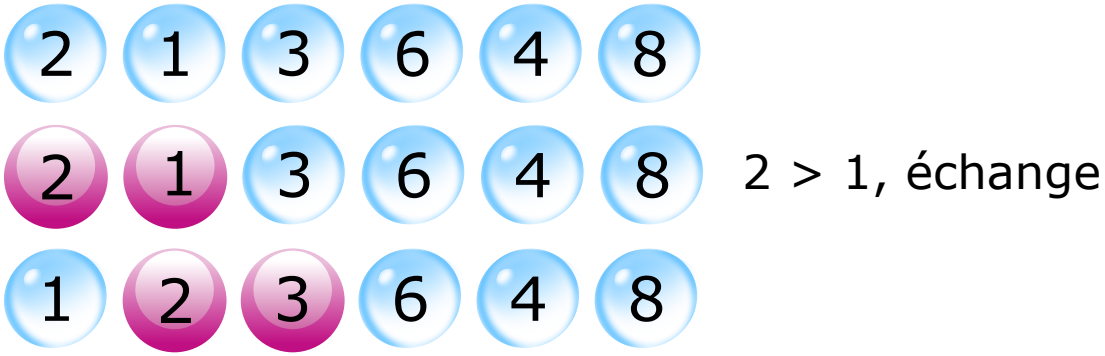
Principe

Un deuxième tour :



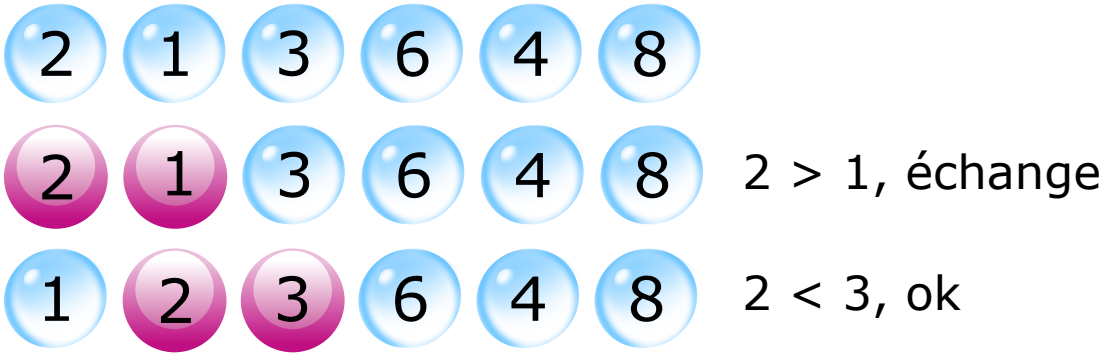
Principe

Un deuxième tour :



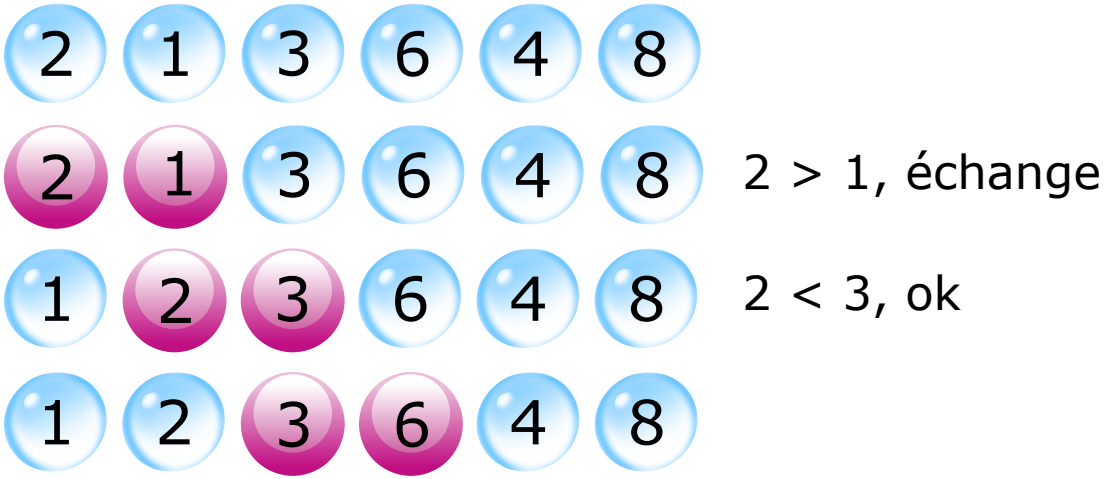
Principe

Un deuxième tour :



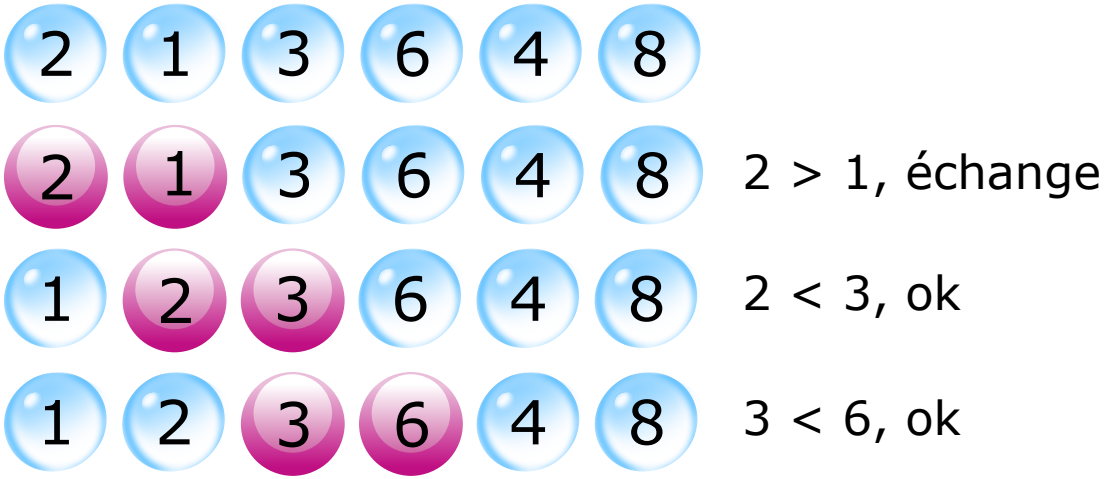
Principe

Un deuxième tour :



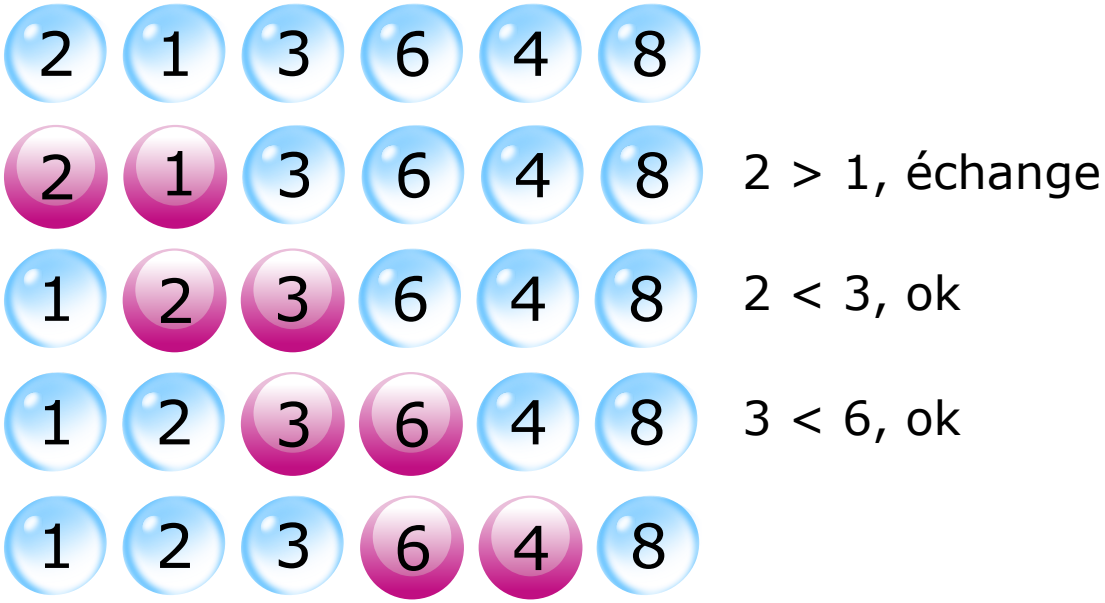
Principe

Un deuxième tour :



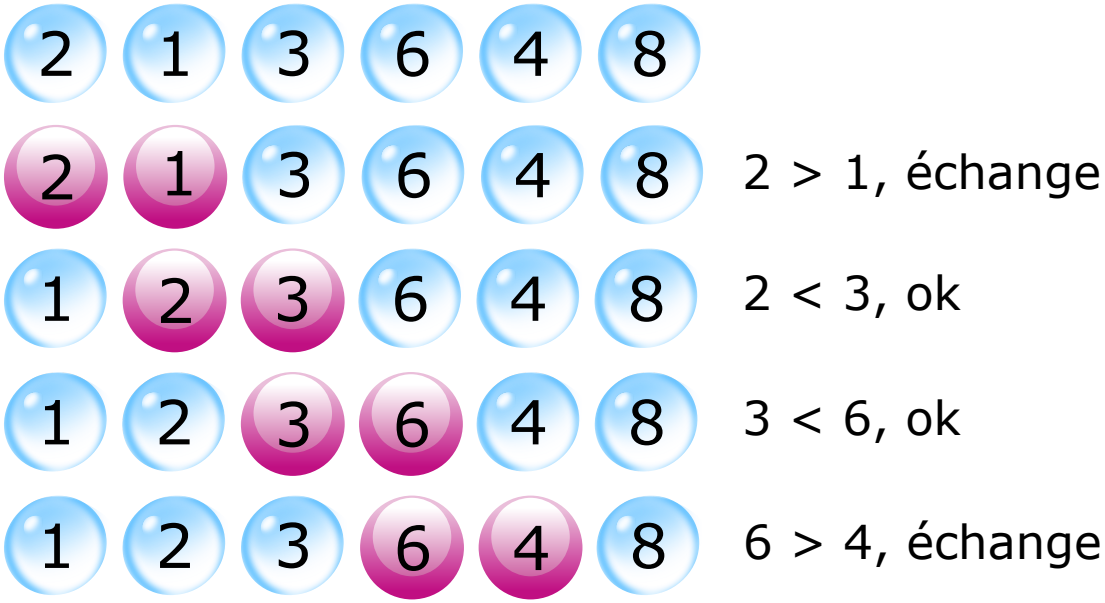
Principe

Un deuxième tour :



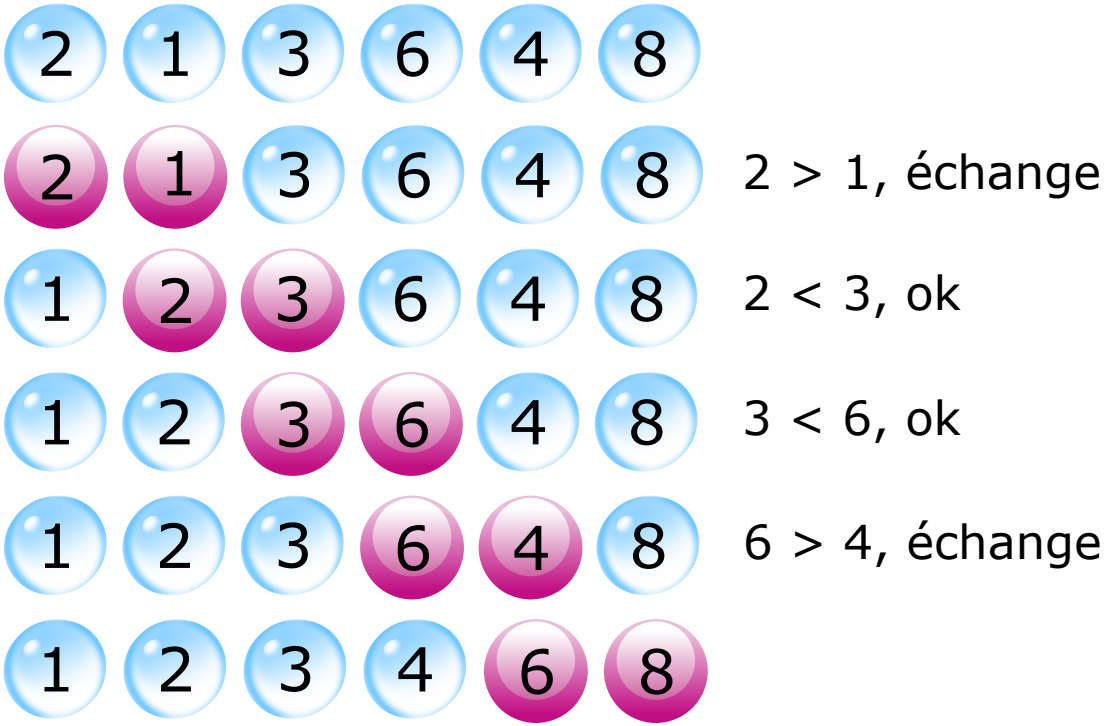
Principe

Un deuxième tour :



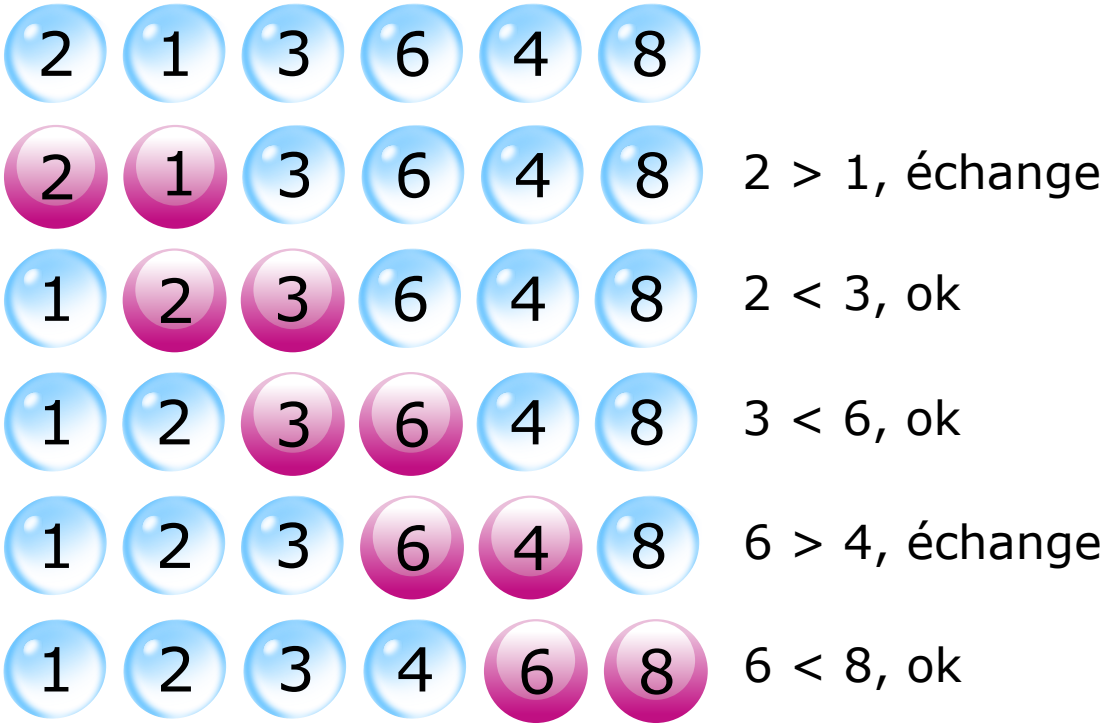
Principe

Un deuxième tour :



Principe

Un deuxième tour :

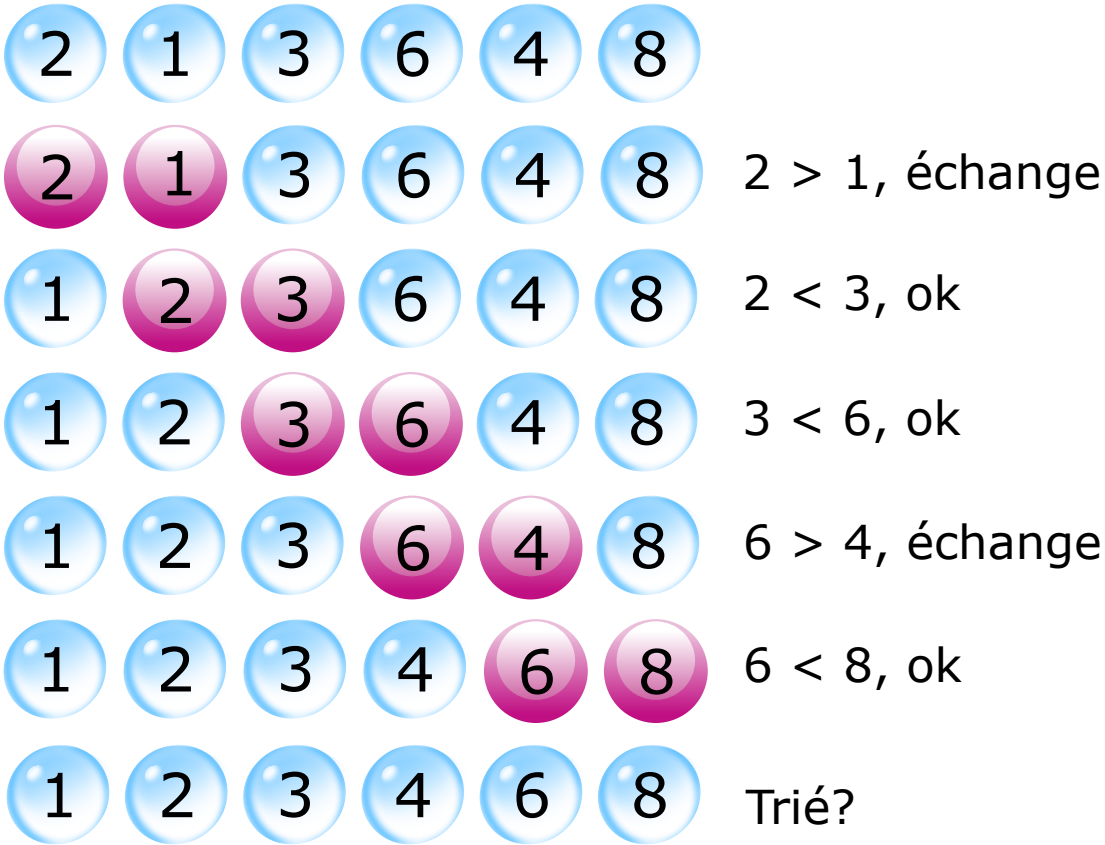


Principe

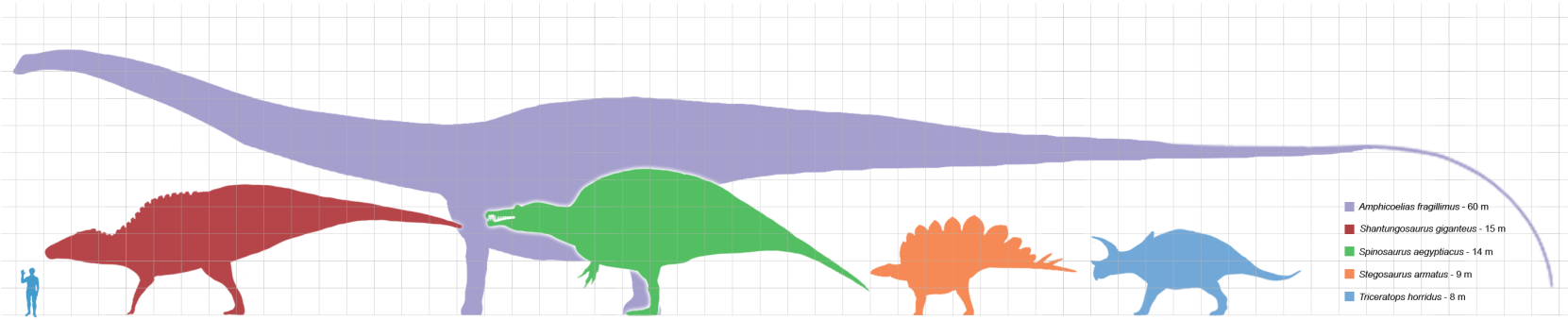
Un deuxième tour :

Principe

Un deuxième tour :



Le tri par sélection



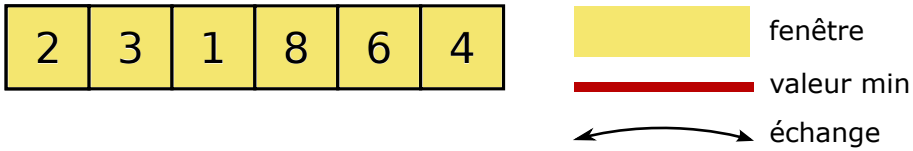
Le tri par sélection

Principe

1. On définit une fenêtre qui couvre l'ensemble du tableau.
2. A chaque étape, on réduit d'une case la fenêtre en allant vers la fin.
4. On échange la première case de la fenêtre avec la valeur minimum de la fenêtre.
3. Quand la fenêtre correspond à l'élément de fin, c'est trié.

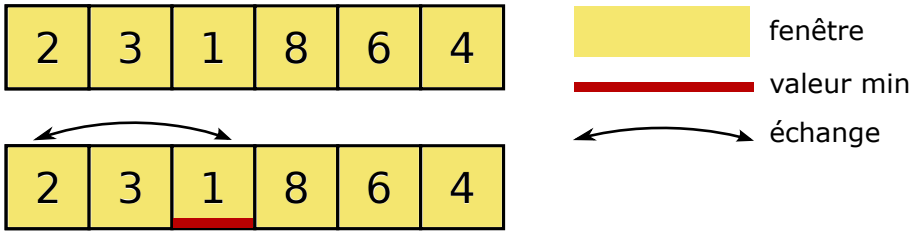
Le tri par sélection

En images



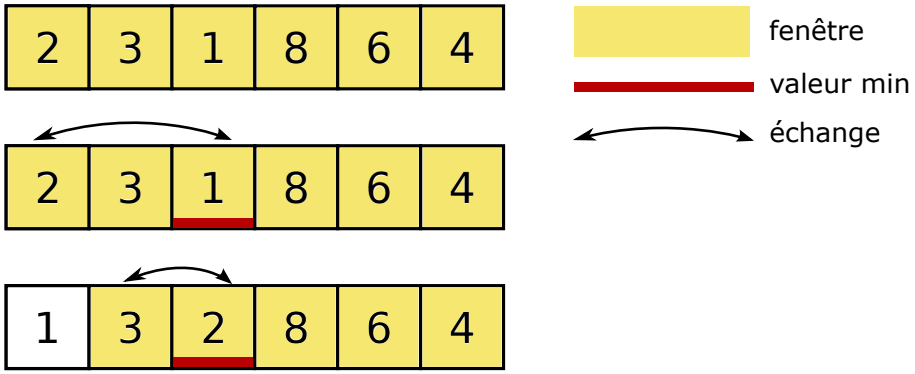
Le tri par sélection

En images



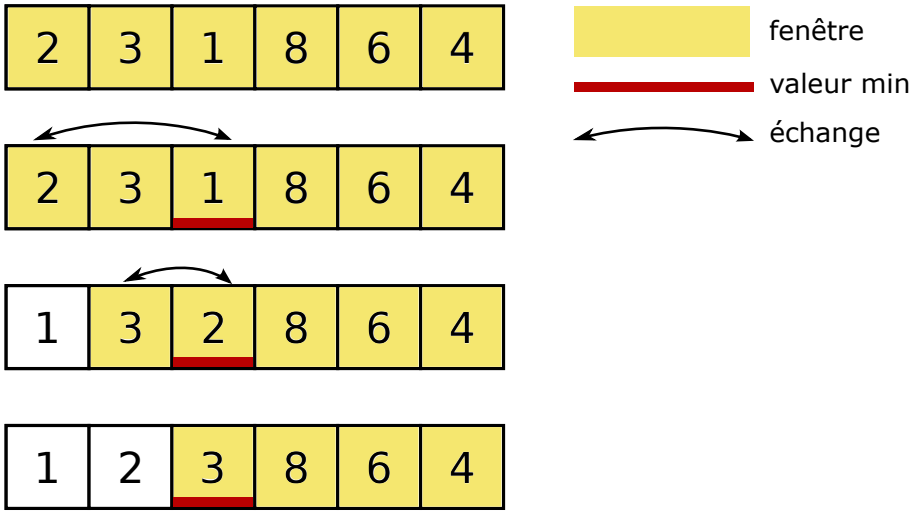
Le tri par sélection

En images



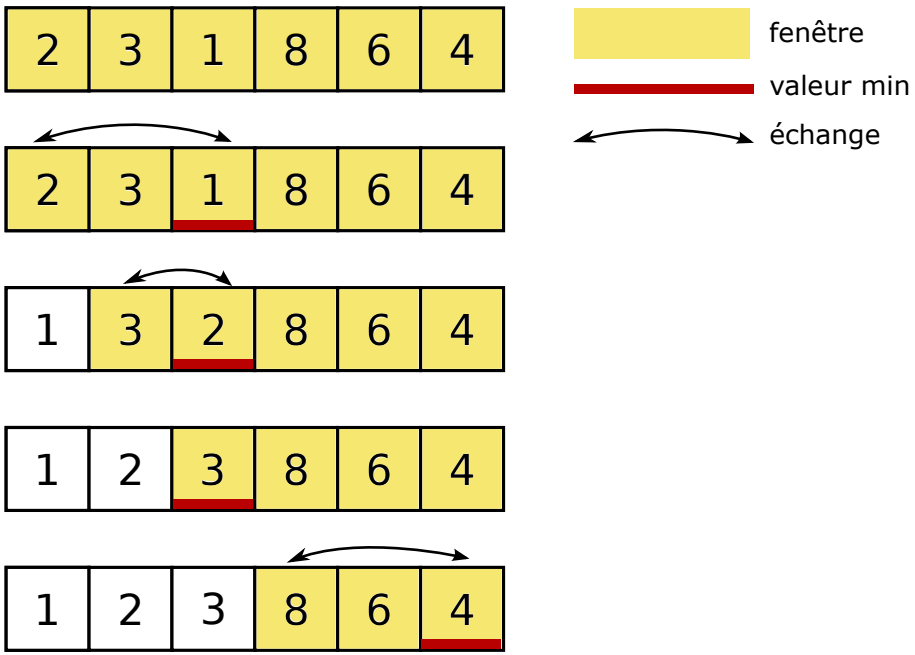
Le tri par sélection

En images



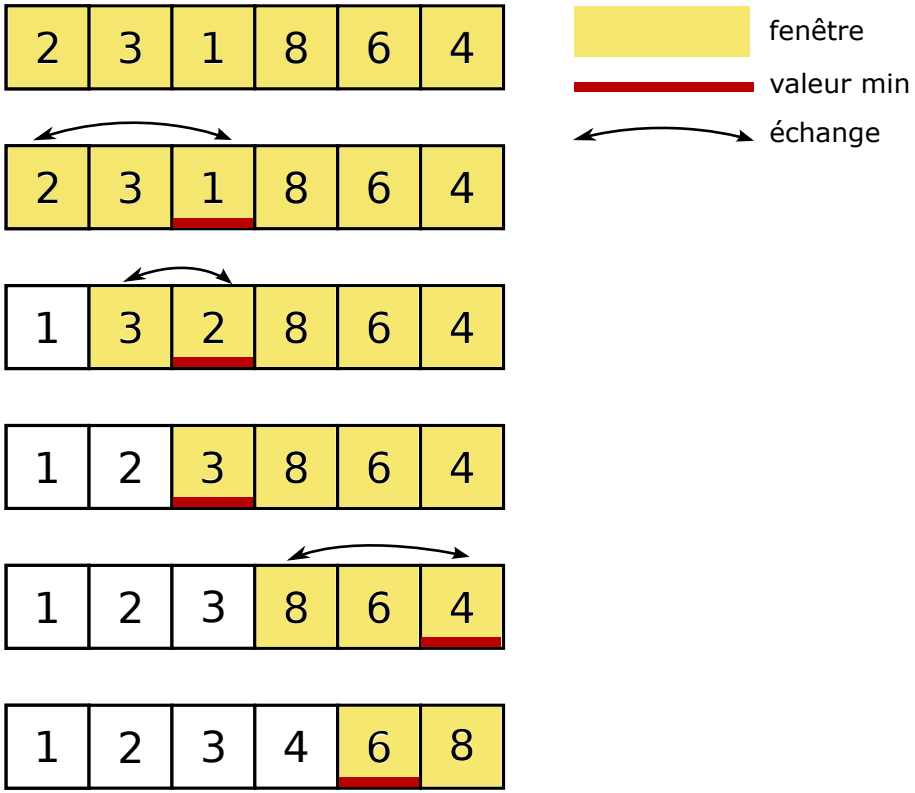
Le tri par sélection

En images



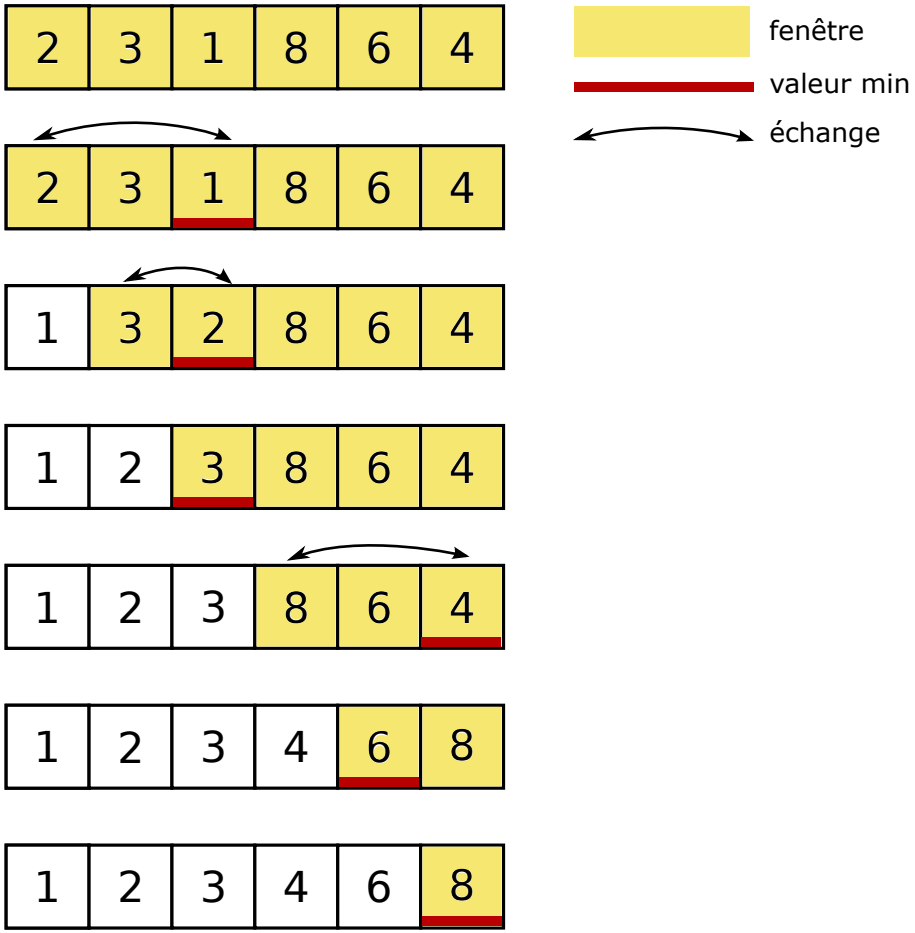
Le tri par sélection

En images

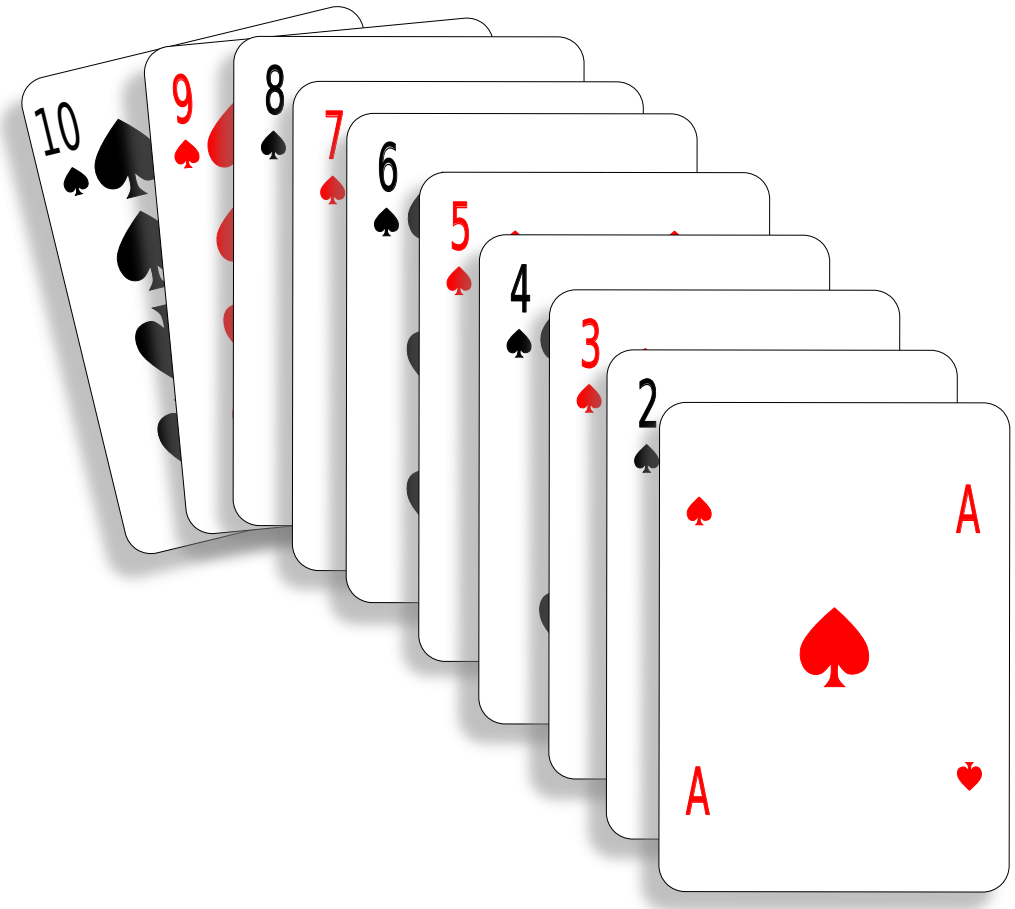


Le tri par sélection

En images



Le tri par insertion



Le tri par insertion

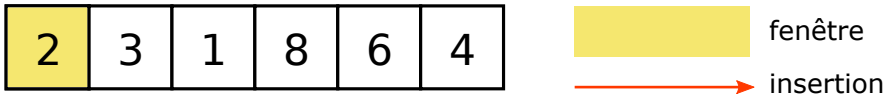
Principe

C'est la manière dont on trie naturellement les cartes que l'on a en main !

1. On définit une fenêtre qui ne contient que le premier élément
2. On parcourt le tableau
3. A chaque étape du parcours, on agrandit la fenêtre d'une case vers la fin
4. On place l'élément courant à sa place dans la fenêtre de manière à ce qu'elle soit triée.

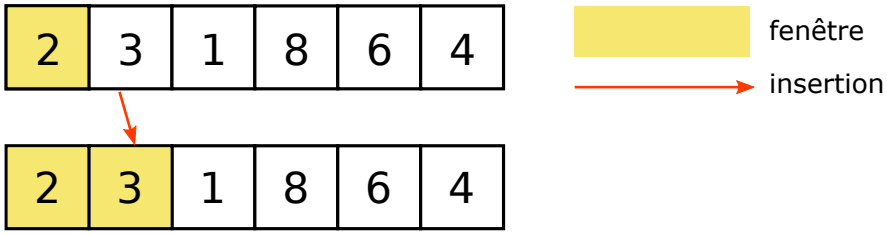
Le tri par insertion

En images



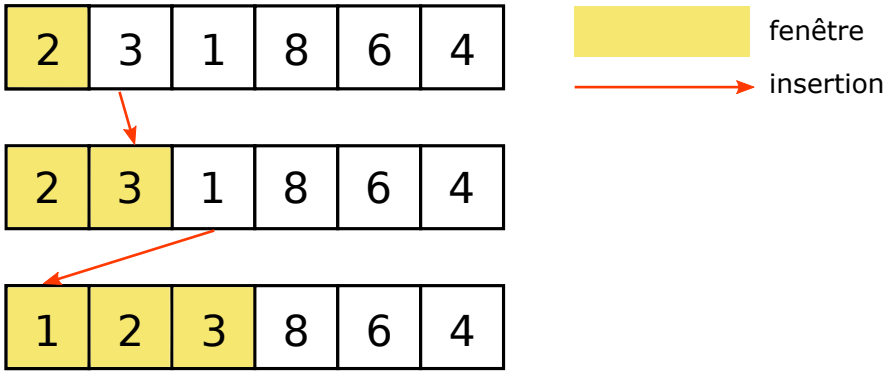
Le tri par insertion

En images



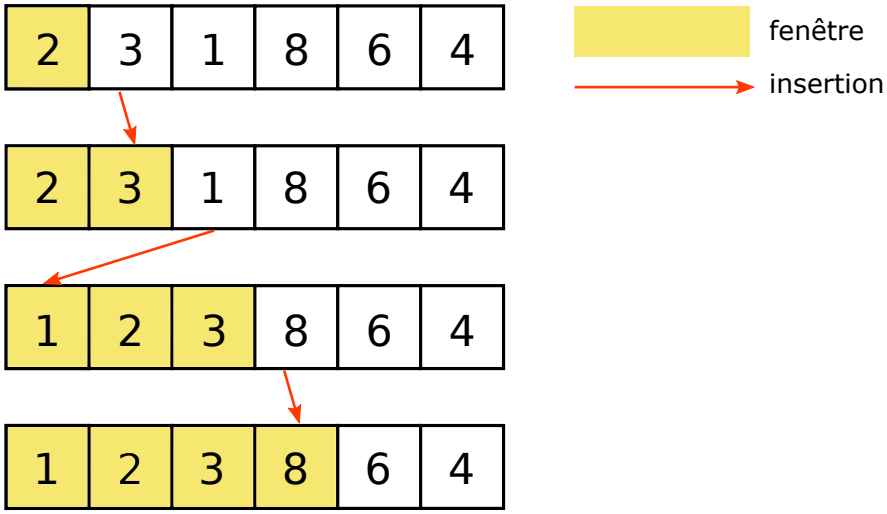
Le tri par insertion

En images



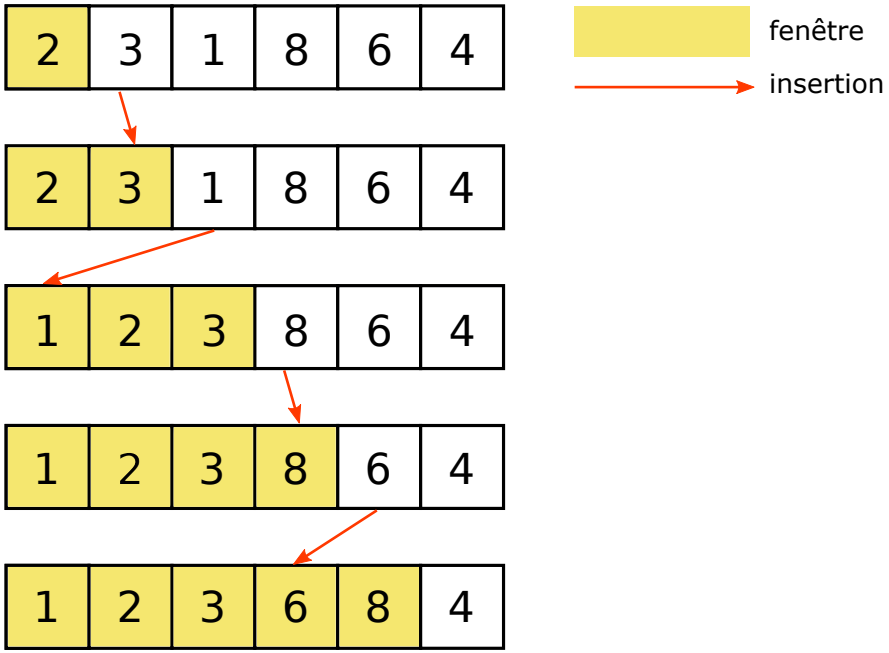
Le tri par insertion

En images



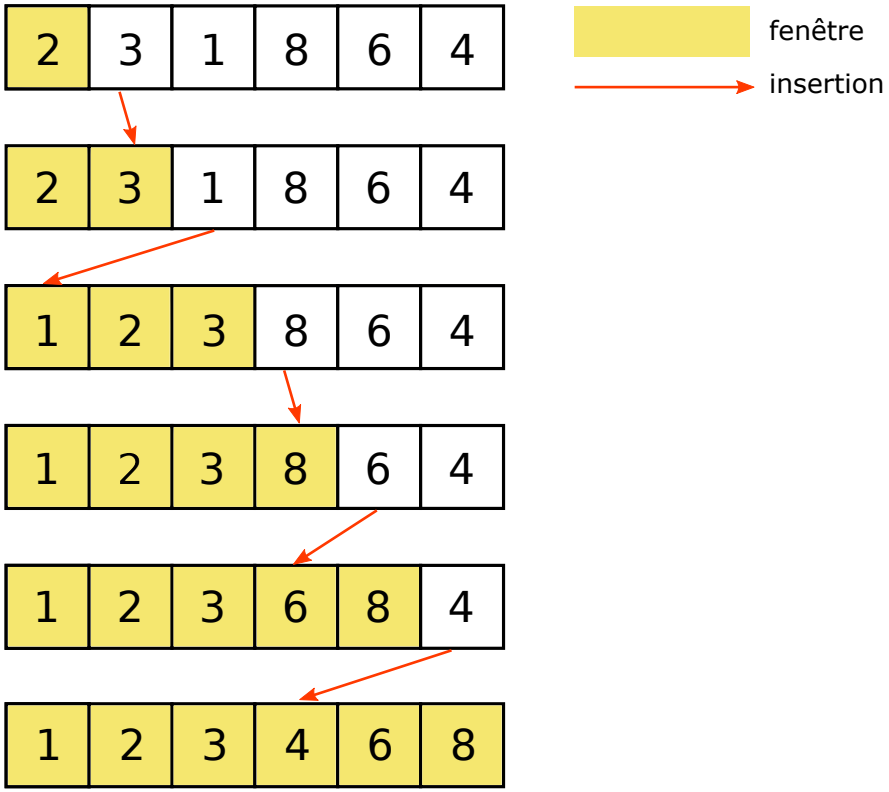
Le tri par insertion

En images



Le tri par insertion

En images



A vous de jouer !

Bilan

Dans ce cours, nous avons vu

- pourquoi trier,
- plusieurs algorithmes de tri (lents),
- des éléments pour étudier leur efficacité.

Bonus

Le Bogosort(ou tri stupide)

Un algorithme de tri diablement inefficace :
$\Rightarrow$ Tant que mon tableau n'est pas trié, je le mélange aléatoirement !

Bonus

Le Bogosort(ou tri stupide)

Un algorithme de tri diablement inefficace :
$\Rightarrow$ Tant que mon tableau n'est pas trié, je le mélange aléatoirement !

```
void bogosort(int [] tab){
  while(!sorted(tab)){
    shuffle(tab)
  }
}
```

Bonus

Le Bogosort(ou tri stupide)

Un algorithme de tri diablement inefficace :
$\Rightarrow$ Tant que mon tableau n'est pas trié, je le mélange aléatoirement !

```
void bogosort(int [] tab){
  while(!sorted(tab)){
    shuffle(tab)
  }
}
```

Implémentez les méthodes `shuffle` et `sorted` !

Bogosort

Solution

```
boolean sorted(int [] tab){
    for(int i = 0; i < tab.length;i++){
        if(tab[i] > tab [i+1]){
        return false;
        }
    }
    return true;
}
```

Bogosort

Solution

```
boolean sorted(int [] tab){
    for(int i = 0; i < tab.length;i++){
        if(tab[i] > tab [i+1]){
        return false;
        }
    }
    return true;
}

void shuffle(int [] tab){
  int idx, idx2;
  for (int i = 0; i < tab.length; i++) {
   idx = ((int) (Math.random() * i.length);
   idx2 = (int) (Math.random() * i.length);
   swap(idx,idx2);
}
```