

# Algorithmique

Amin NAIRI <[anairi@esgi.fr](mailto:anairi@esgi.fr)>

# Séance 1 — Introduction à l'algorithmique

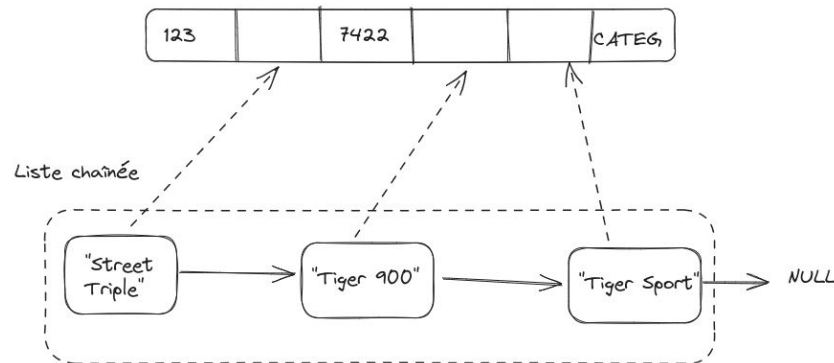
- Présentation du cours, objectifs & attentes
- Introduction aux algorithmes et à leur importance dans la résolution de problèmes
- Notions de bases : données, variables, instructions, structures de contrôles
- Exemples simples d'algorithmes
- Projet pédagogique

# Séance 2 — Structures de données

- Tableaux
- Listes chaînées
- Piles
- Files
- Avantages & inconvénients de chaque structures de données
- Manipulation de structures de données en algorithmique

# Séance 2 — Liste chaînée

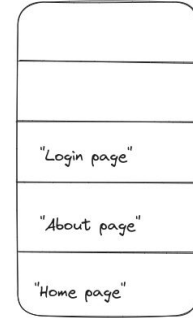
- Les tableaux classiques stockent des données de manière contigüe
- Les Listes Chaînées sont utilisées pour pouvoir stocker des données de manière non-contigüe
- Idéal pour pouvoir optimiser l'espace en mémoire utilisé par une structure de données
- Obligé de se déplacer dans toute la liste pour chercher un élément à un index spécifique, contrairement à un tableau



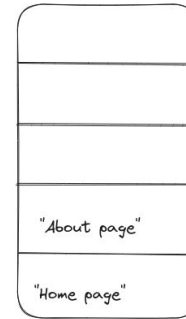
# Séance 2 — Pile (Stack)

- Idéal pour stocker des choses qui peuvent être corrigées comme des changements sur un éditeur, ou un historique de navigation
- Le dernier élément inséré est le premier élément sorti (LIFO, Last In First Out)
- Repose sur une liste chaînée dans son implémentation

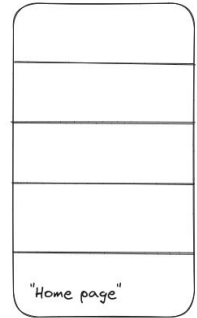
```
$stack->push("Home page");  
$stack->push("About page");  
$stack->push("Login page");
```



```
$stack->pop();
```

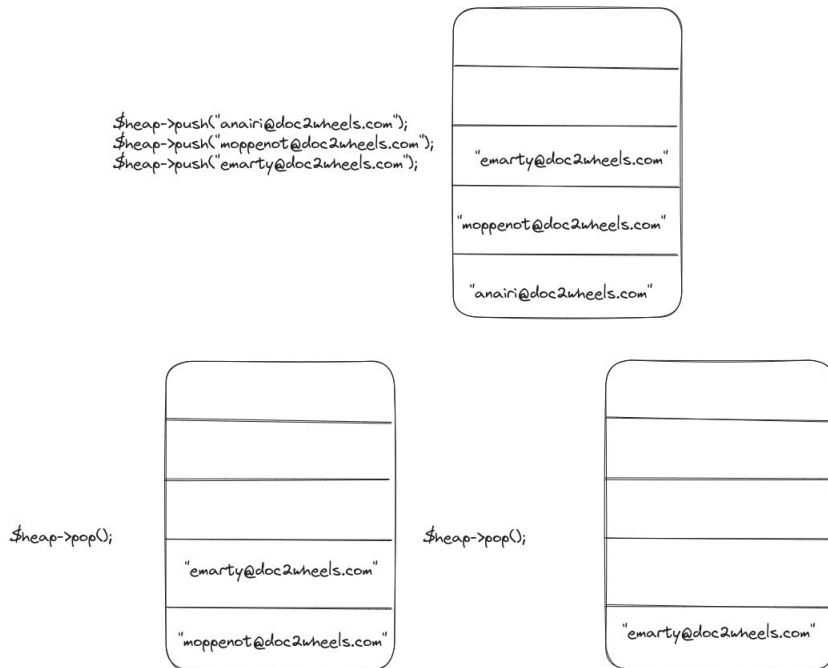


```
$stack->pop();
```



# Séance 2 — File (Heap)

- Idéal pour stocker des queues comme pour pouvoir envoyer des emails de manière asynchrone plus tard par exemple
- Le premier élément inséré est le premier élément sorti (FIFO, First In First Out)
- Repose sur une liste chaînée dans son implémentation

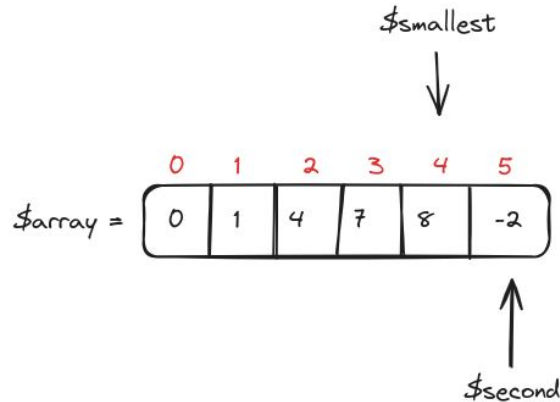


# Séance 3 — Algorithmes de recherche et de tri

- Algorithmes de recherche linéaire et binaire
- Algorithmes de tri : sélection, insertion, à bulles
- Complexité des algorithmes de recherche et de tri
- Contrôle continu #1

# Séance 3 — Tri par sélection

$\$first = 4$



Algorithme de tri par sélection

Tour de boucle

Est-ce que à chaque tour de boucle  
 $\$smallest$  est toujours plus petit  
que  $\$second$  (index)

Comparaison

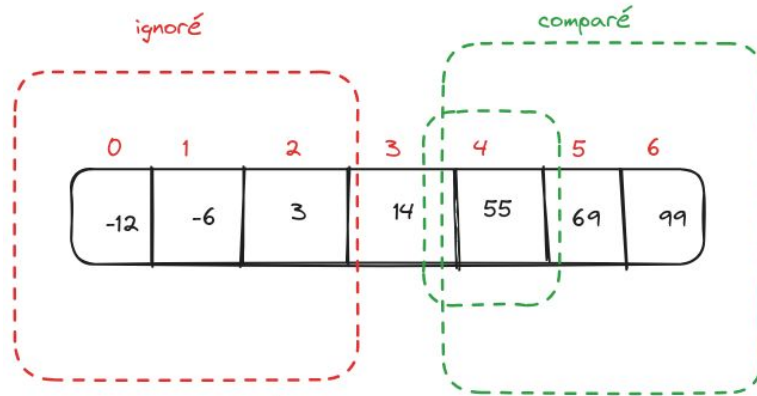
```
if ( $\$array[\$second] < \$array[\$smallest]$ )  
     $\$smallest = \$second;$ 
```

Echange

```
 $\$temporary = \$array[\$first];$   
 $\$array[\$first] = \$array[\$smallest];$   
 $\$array[\$smallest] = \$temporary;$ 
```



# Séance 3 — Recherche binaire



`$itemsCount = 7;`

`$middleIndex = floor($itemsCount / 2);`

`$searchValue = 55;`

```
if ($items[$middleIndex] === $searchValue)
    return $middleIndex;
```

Recherche binaire  
« diviser pour mieux régner »

VS

Recherche linéaire  
boucle sur tous les éléments