

# Algorithmique Appliquée

BTS SIO SISR

Résolution de problèmes  
classiques

# Plan

- Listes chaînées
- Queue et FIFO
- Stack et LIFO
- Comparaison entre FIFO et LIFO
- Rappels sur la théorie des ensembles
- Rappels sur le calcul matriciel avancé

# Listes chaînées

Tableau à taille fixe dans d'autres langages de programmation  
Vecteur de taille dynamique dans d'autres langages (réallocation mémoire)  
Liste chaînée : représentation graphique  
Montrer graphiquement l'insertion, suppression, recherche...  
Version immutable d'une liste chaînée : insertion au début uniquement.  
Avantage d'une liste chaînée : insertion au début ou à la fin très rapide.  
Inconvénients d'une liste chaînée : pas d'indexation et mémoire éparse en contradiction avec les architectures de CPU modernes (cache miss, etc.)  
Comparaison avec une `list` Python : c'est plutôt en vecteur pour des raisons de performance mais en vrai je ne crois pas que les détails d'implémentation fassent parti de la spec (@TBC)

# TP : Manipulation d'une liste chaînée

# **TP : Manipulation d'une liste chaînée**

[Lien vers le sujet de TP.](#)



# Queue et FIFO

First-In, First-Out

Premier entré, premier servi, comme au restau  
Queue == conteneur spécialisé pour résoudre cela (file  
d'attente)

Représentation graphique

Trouver une vidéo qui montre l'empilement et le  
dépilement des messages

Utilisation de `list`

Structure de données plus adaptée : `dequeue`

Montrer des exemples



# Stack et LIFO

Last-In, First-Out

Dernier entré, premier servi, comme un jeu  
d'empilement avec une base ou une callstack  
Stack == conteneur spécialisé pour résoudre cela

Représentation graphique

Trouver une vidéo qui montre l'empilement et le  
dépilement des messages

Utilisation de `list`

Structure de données plus adaptée : `LifoQueue`

Montrer des exemples

# Comparaison entre FIFO et LIFO

Comparaison graphique  
Queue plus souvent utilisée en pratique  
Répondent à des besoins différents

# TP : Queues de messages simple

# TP : Queues de messages simple

[Lien vers le sujet de DM.](#)



# Rappels sur la théorie des ensembles

Union, intersection, exclusion

En prévision du DM 03 et de l'examen final

Intérêt : on manipule régulièrement des ensembles de données (data mining, data science)

Rappels rapides avec des patatoïdes

Exemple avec entiers et nombres réels

Exemple (différents) avec les `int` et les `float`

Structure de données pour représenter un ensemble :

`set`

Ensembles disjoints, inclus, intersection

Union, exclusion

# Rappels sur le calcul matriciel avancé

Diagonalisation, inversion, méthode de Gauss

En prévision du DM 03 et de l'examen final

Intérêt : pour faire des jeux vidéos, du CAD/FEA ou  
des systèmes de guidage de missiles

Autre intérêt : résoudre les systèmes d'équations  
simples

Transformée d'une matrice

Diagonalisation de matrice

Déterminant et Co-matrice

Inversion de matrice avec co-matrice

Matrices de transformation 2D et 3D

Inversion d'une matrice de transformation

Poser un système d'équation au format matriciel

Résolution avec la méthode de Gauss

# Devoir à la Maison 03

# **DM : Retours sur les fonctions et le débogage**

[\*\*Lien vers le sujet de DM.\*\*](#)