XWIKI NATIS

Algorithme #2

Michael

XWIKI NATIS

XDNATIS

Compétence demandée :
Comprendre les concepts algorithmiques intermédiaires

XKNATIS

1. Récursivité
2. Tri
3. DFS
4. Compression
5. Décisionnel

XWIKI SAS

XWIKI NATIS

1. Récursivité
2. Tri
3. DFS
4. Compression
5. Décisionnel

XWIKI NATIS

Récursivité

XWIKI NATIS

Qu’est-ce que la récurrence ?

XEBIANATIS

Un algorithme récursif est un algorithme qui résout un problème en calculant des solutions d'instances plus petites du même problème. L'approche récursive est un des concepts de base en informatique.

XWIKI NATIS

Un algorithme récursif peut être considéré comme une fonction qui s'appelle elle-même ! Cela crée donc une boucle (implicite).

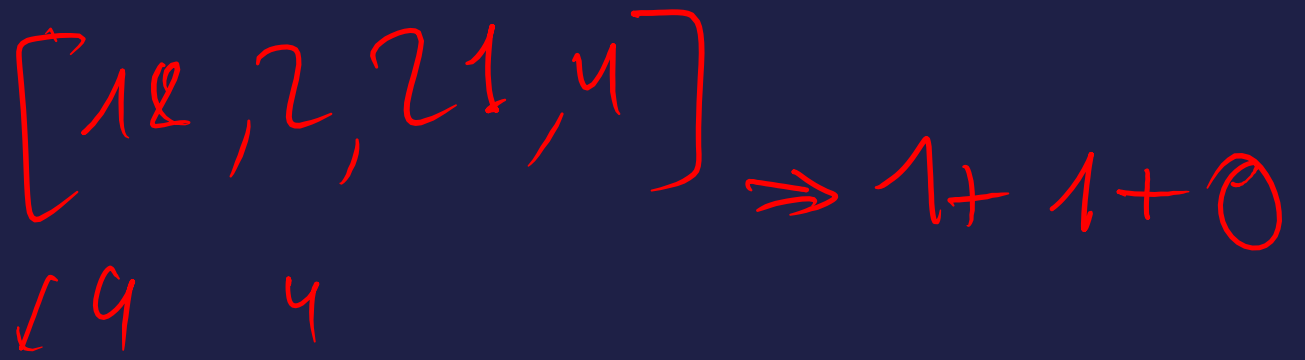
La récursion est une alternative élégante à la boucle.

XKNATIS

Pour qu'un algorithme puisse être appelé, il faut qu'il soit nommé et encapsulé dans une procédure ou une fonction.

```
@Fonction compter @Entrée tab, i, longueur
    @DebutBloc
    @Si i >= longueur
        @DebutBloc
        @Retourner 0
        @FinBloc

    @Si tab[i] >= 18
        @DebutBloc
        @Retourner 1 + compter @Avec tab, i + 1, longueur
        @FinBloc
    @Sinon
        @DebutBloc
        @Retourner compter @Avec tab, i + 1, longueur
        @FinBloc

    @FinBloc
```

XNATIS



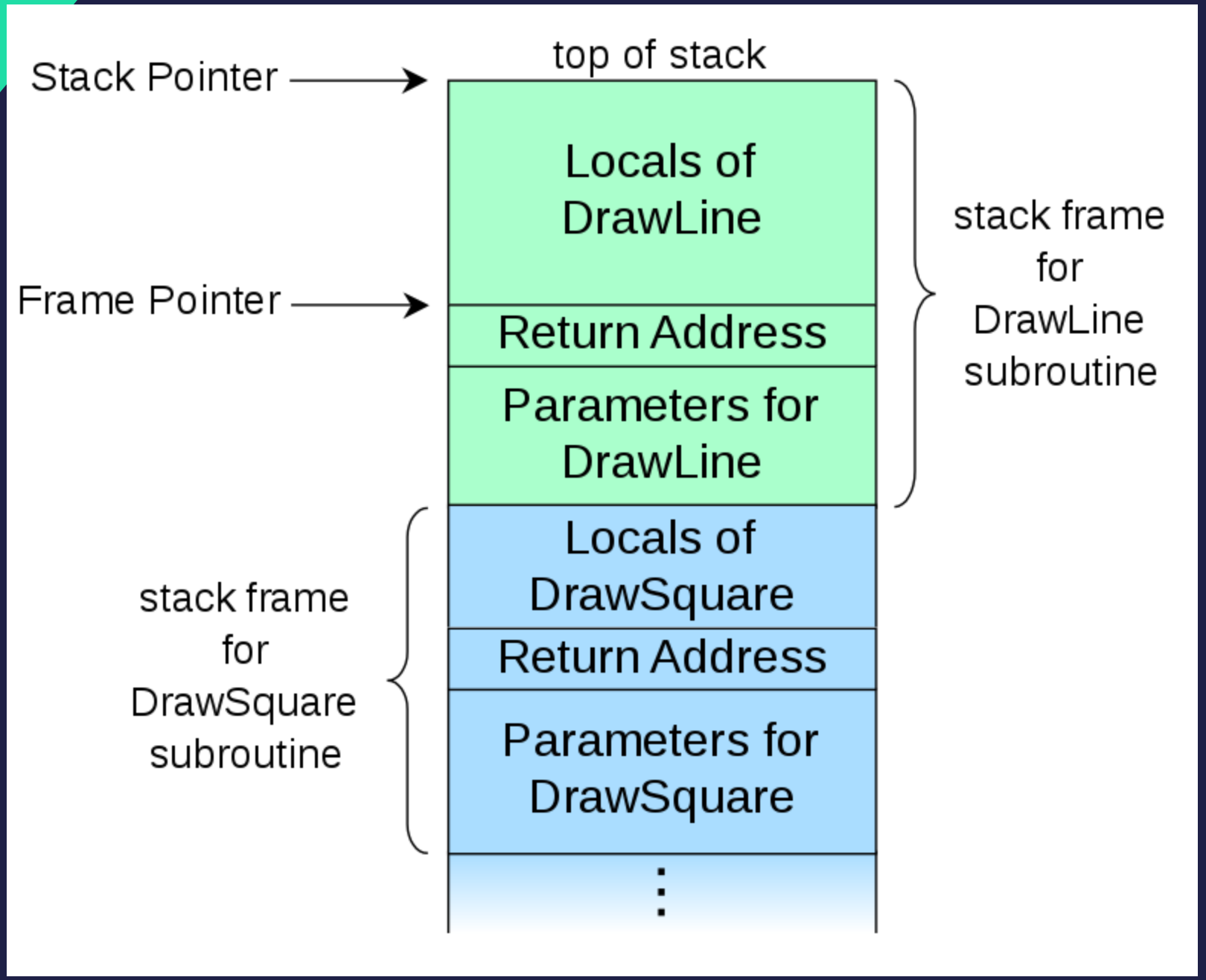

```
@Fonction compter @Entrée tab, i, longueur
    @DebutBloc
    @Si i >= longueur
        @DebutBloc
        @Retourner 0
        @FinBloc

    @Si tab[i] >= 18
        @DebutBloc
        @Retourner 1 + compter @Avec tab, i + 1, longueur
        @FinBloc
    @Sinon
        @DebutBloc
        @Retourner compter @Avec tab, i + 1, longueur
        @FinBloc

    @FinBloc
```

XEBIA NATIS

En informatique, la pile d'exécution (souvent abrégée en la pile ; en anglais, call stack) est une structure de données de type pile qui sert à enregistrer des informations au sujet des fonctions actives dans un programme informatique.

XNATIS

```
@Fonction DrawLine @Entrée x1, y1, x2, y2
    @DebutBloc
    // Dessiner une ligne de (x1, y1) à (x2, y2)
    @FinBloc

@Fonction DrawRectangle
    @DebutBloc

    DrawLine @Avec 10, 10, 100, 10
    DrawLine @Avec 100, 10, 100, 100
    DrawLine @Avec 100, 100, 10, 100
    DrawLine @Avec 10, 100, 10, 10
    @FinBloc
```


top of stack
Stack Pointer
Frame Pointer
Locals of DrawLine
Return Address
Parameters for DrawLine
stack frame for DrawLine subroutine
Locals of DrawSquare
Return Address
Parameters for DrawSquare
stack frame for DrawSquare subroutine

XWIKI NATIS

La récursion prend plus d'espace en RAM qu'un simple boucle. L'espace occupé est proportionnel au nombre d'itérations.

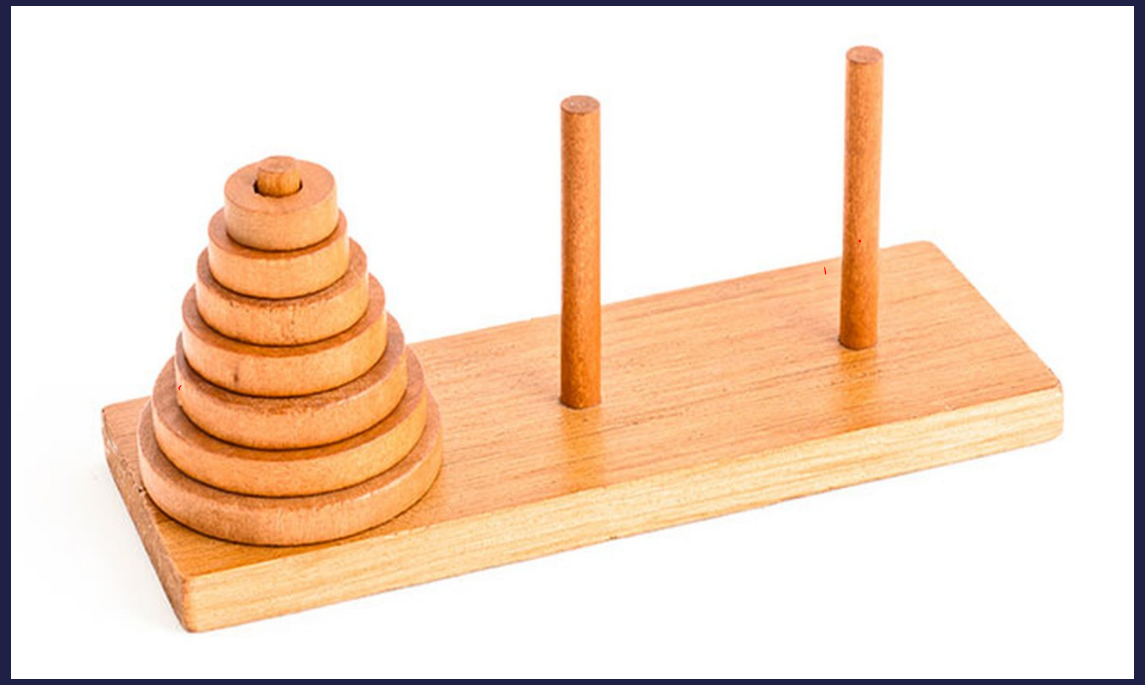
XWIKI NATIS

Comment supprimer la récursivité dans un algorithme ?

XWIKI SAS

Tours de Hanoi

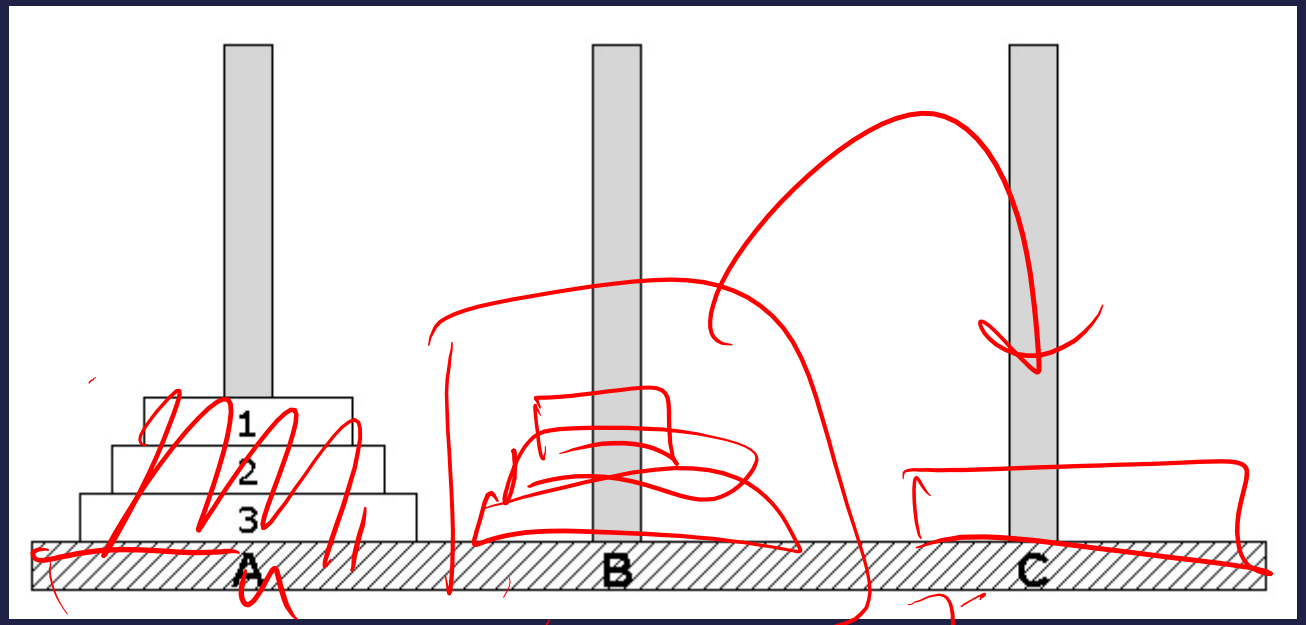
XWNATIS



- On ne peut déplacer plus d'un disque à la fois
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide

XWIKI SAS

XEBIA NATIS



XWIKI SAS

XWIKI SAS

XWIKI SAS

XWIKI SAS

XEBIANATIS

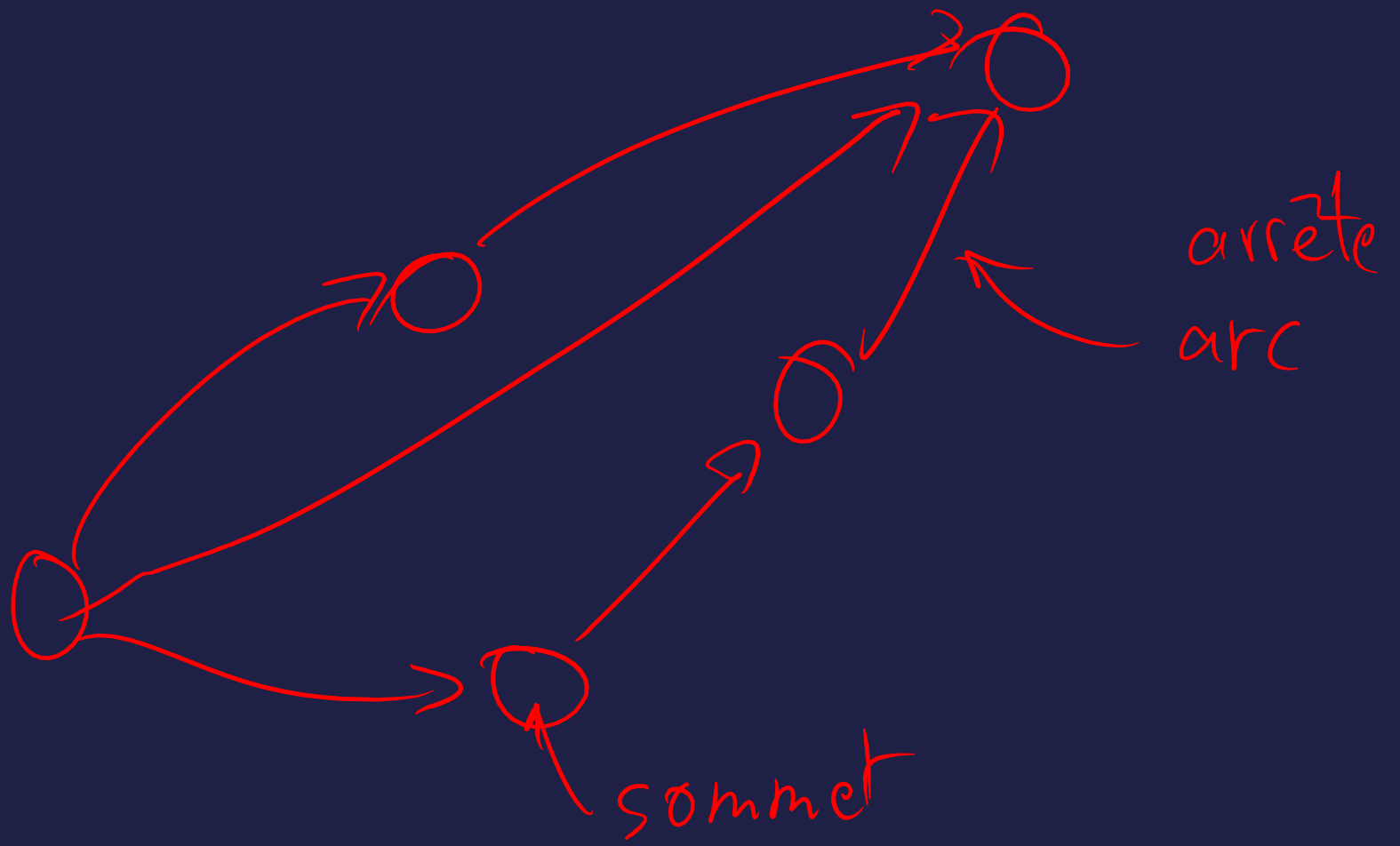
```
@Fonction deplacer @Entrée disques, pileDepart, pileDestination, pileBuffer
    @DebutBloc

    @Si disques vide
        @DebutBloc
        @Retourner
        @FinBloc

    deplacer @Avec disques[0..n-2], pileDepart, pileBuffer, pileDestination
    Push disques[n-1] sur pileDestination
    déplacer @Avec disques[0..n-2], pileBuffer, pileDestination, pileDepart

    @FinBloc
```

Théorie des graphes

XWIKI NATIS

La théorie des graphes est la discipline mathématique et informatique qui étudie les graphes, lesquels sont des modèles abstraits de dessins de réseaux reliant des objets.

XWIKI NATIS



XWIKI NATIS

Ces modèles sont constitués par la donnée de sommets (aussi appelés nœuds ou points, en référence aux polyèdres), et d'arêtes (aussi appelées liens ou lignes) entre ces sommets ; ces arêtes sont parfois non-symétriques (les graphes sont alors dits orientés) et sont appelées des flèches ou des arcs.

XMNATIS

En théorie des graphes, l'algorithme de Dijkstra sert à résoudre le problème du plus court chemin. Il permet, par exemple, de déterminer un plus court chemin pour se rendre d'une ville à une autre connaissant le réseau routier d'une région.

XDNATIS

$$C_{Nantes \to Paris} = \min\left[\begin{array}{l} 1 + C_{Nantes\ Angers} \\ 3 + C_{Nantes\ Tours} \end{array}\right]$$

XWIKI NATIS

L'algorithme de Bellman-Ford, aussi appelé algorithme de Bellman–Ford–Moore, est un algorithme qui calcule des plus courts chemins depuis un sommet source donné dans un graphe orienté pondéré.

XWIKI NATIS

Contrairement à l'algorithme de Dijkstra, l'algorithme de Bellman-Ford autorise la présence de certains arcs de poids négatif et permet de détecter l'existence d'un circuit absorbant, c'est-à-dire de poids total strictement négatif, accessible depuis le sommet source.

XWIKI SAS

XWIKI SAS

XKNATIS

1. Récursivité
2. Tri
3. DFS
4. Compression
5. Décisionnel

XWIKI NATIS

Tri

XWIKI NATIS

Pourquoi trier des données/valeurs ?

XWIKI NATIS

Un algorithme de tri est, en informatique ou en mathématiques, un algorithme qui permet d'organiser une collection d'objets selon une relation d'ordre déterminée.

XEBIANATIS

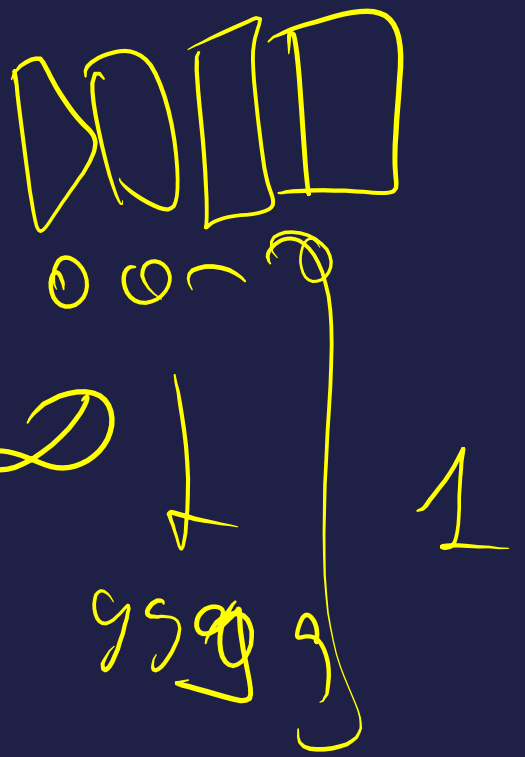
Nom	Cas optimal	Cas moyen	Pire des cas	Complexité spatiale	Stable
Tri rapide	$n \log n$	$n \log n$	n^2	$\log n$ en moyenne, n dans le pire des cas ; variante de Sedgewick : $\log n$ dans le pire des cas	Non
Tri fusion	$n \log n$	$n \log n$	$n \log n$	n	Oui
Tri par tas	$n \log n$	$n \log n$	$n \log n$	1	Non
Tri par insertion	n	n^2	n^2	1	Oui
Introsort	$n \log n$	$n \log n$	$n \log n$	$\log n$	Non
Tri par sélection	n^2	n^2	n^2	1	Non
Timsort	n	$n \log n$	$n \log n$	n	Oui
Tri de Shell	n	$n \log^2 n$ ou $n^{3/2}$	$n \log^2 n$pour la meilleure suite d'espacements connue	1	Non
Tri à bulles	n	n^2	n^2	1	Oui
Tri arborescent	$n \log n$	$n \log n$	$n \log n$ (arbre équilibré)	n	Oui
Smoothsort	n	$n \log n$	$n \log n$	1	Non
Tri cocktail	n	n^2	n^2	1	Oui
Tri à peigne	n	$n \log n$	n^2	1	Non
Tri pair-impair	n	n^2	n^2	1	Oui

XWIKI NATIS

Quelle complexité est meilleure ?

XKNATIS

Comportement asymptotique



$$O(1) < O(\log n) < O(n) < O(n \log(n)) < O(n*n) < O(\exp(n))$$

Plus rapide

Plus lent

Problème NP-complet

XWIKI NATIS

1. Tri à bulles
2. Tri casier
3. Tri fusion

XWIKI NATIS

Tri à bulles

XWIKI NATIS

Tri à bulles

Il consiste à faire remonter le maximum en bout de tableau par permutations.

Il doit son nom au fait qu'il déplace rapidement les plus grands éléments en fin de tableau, comme des bulles d'air qui remonteraient rapidement à la surface d'un liquide.

XNATIS

Tri casier

XKNATIS

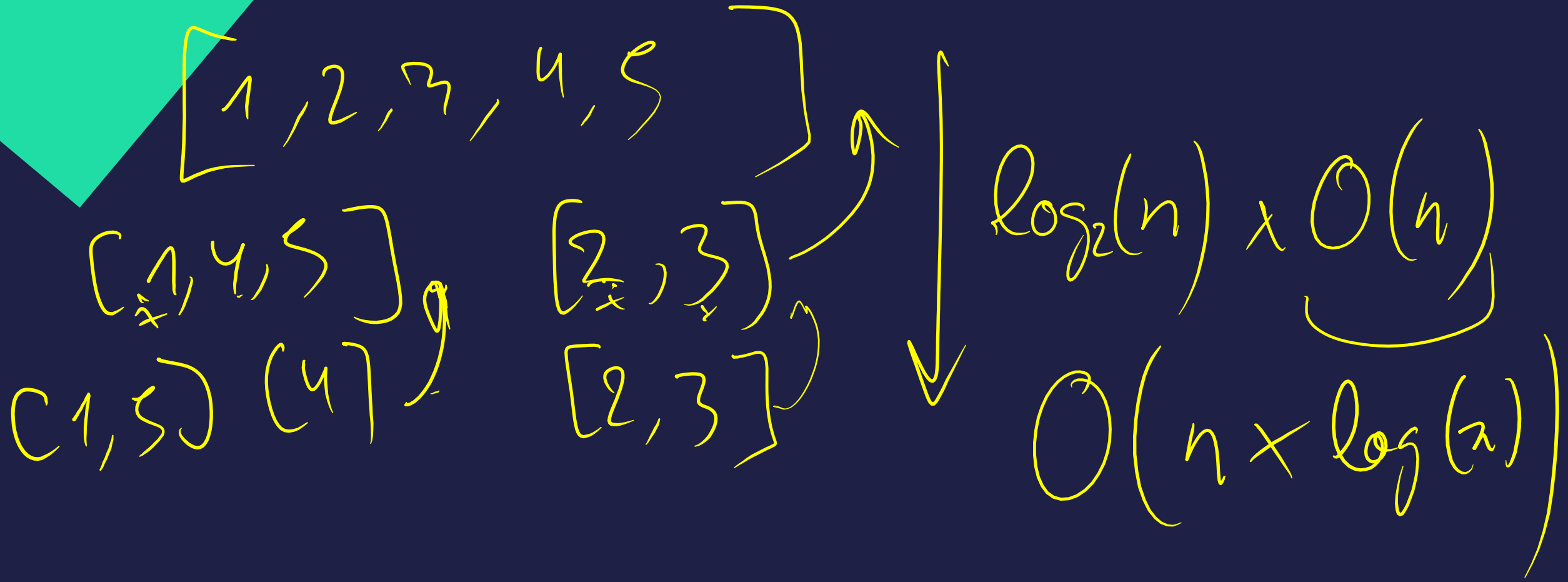
Tri casier

Le tri casier est un tri sans comparaison. Il consiste à distribuer les valeurs dans des casiers et les empiler par après.

Il est aussi appelé tri comptage.

XEBIA NATIS

Tri fusion



XWIKI NATIS

Tri fusion

Le tri fusion, ou merge sort, est un algorithme de tri par comparaison stable.

Sa complexité temporelle pour une entrée de taille n est de l'ordre de n log n, ce qui est asymptotiquement optimal.

XWIKI NATIS

Comment est fait le tri des langages interprétés ?

XNATIS

Nom	Cas optimal	Cas moyen	Pire des cas	Complexité spatiale	Stable
Tri rapide	$n \log n$	$n \log n$	n^2	$\log n$ en moyenne, n dans le pire des cas ; variante de Sedgewick : $\log n$ dans le pire des cas	Non
Tri fusion	$n \log n$	$n \log n$	$n \log n$	n	Oui
Tri par tas	$n \log n$	$n \log n$	$n \log n$	1	Non
Tri par insertion	n	n^2	n^2	1	Oui
Introsort	$n \log n$	$n \log n$	$n \log n$	$\log n$	Non
Tri par sélection	n^2	n^2	n^2	1	Non
Timsort	n	$n \log n$	$n \log n$	n	Oui
Tri de Shell	n	$n \log^2 n$ ou $n^{3/2}$	$n \log^2 n$pour la meilleure suite d'espacements connue	1	Non
Tri à bulles	n	n^2	n^2	1	Oui
Tri arborescent	$n \log n$	$n \log n$	$n \log n$ (arbre équilibré)	n	Oui
Smoothsort	n	$n \log n$	$n \log n$	1	Non
Tri cocktail	n	n^2	n^2	1	Oui
Tri à peigne	n	$n \log n$	n^2	1	Non
Tri pair-impair	n	n^2	n^2	1	Oui

XEBIA NATIS

Quelle est la meilleure complexité en pire des cas pour les algorithmes de tri basés sur des comparaisons ?

XWIKI SAS

XWIKI SAS

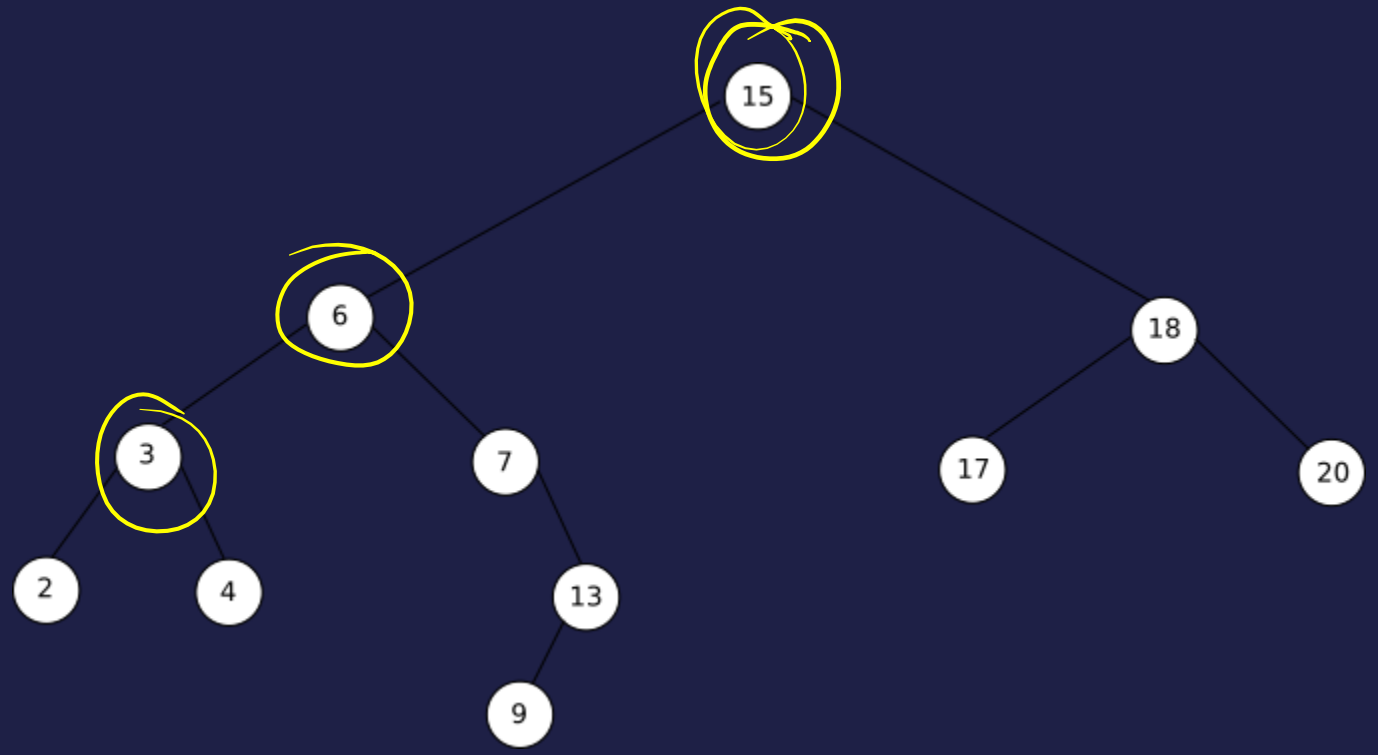
XKNATIS

1. Récursivité
2. Tri
3. DFS
4. Compression
5. Décisionnel

XWIKI NATIS

DFS

XEBIANATIS



XWINATIS

DFS + BST

Le DFS mené sur un BST produit une liste triée

Binary Search Tree

XWIKI NATIS

DFS non récursif ?

XEBIA NATIS

DFS

Nous pouvons supprimer la récursivité et obtenir un algorithme non récursif en utilisant une pile (une structure LIFO) !

Implémentation récursive

Implémentation en pile (LIFO)

XWIKI SAS

XWIKI SAS

XKNATIS

1. Récursivité
2. Tri
3. DFS
4. Compression
5. Décisionnel

XDNATIS

COMPRESSION

Algorithme de Huffman

XWIKI

Le codage de Huffman est un algorithme de compression de données sans perte. Le codage de Huffman utilise un code à longueur variable pour représenter un symbole de la source (par exemple un caractère dans un fichier).

XONATIS

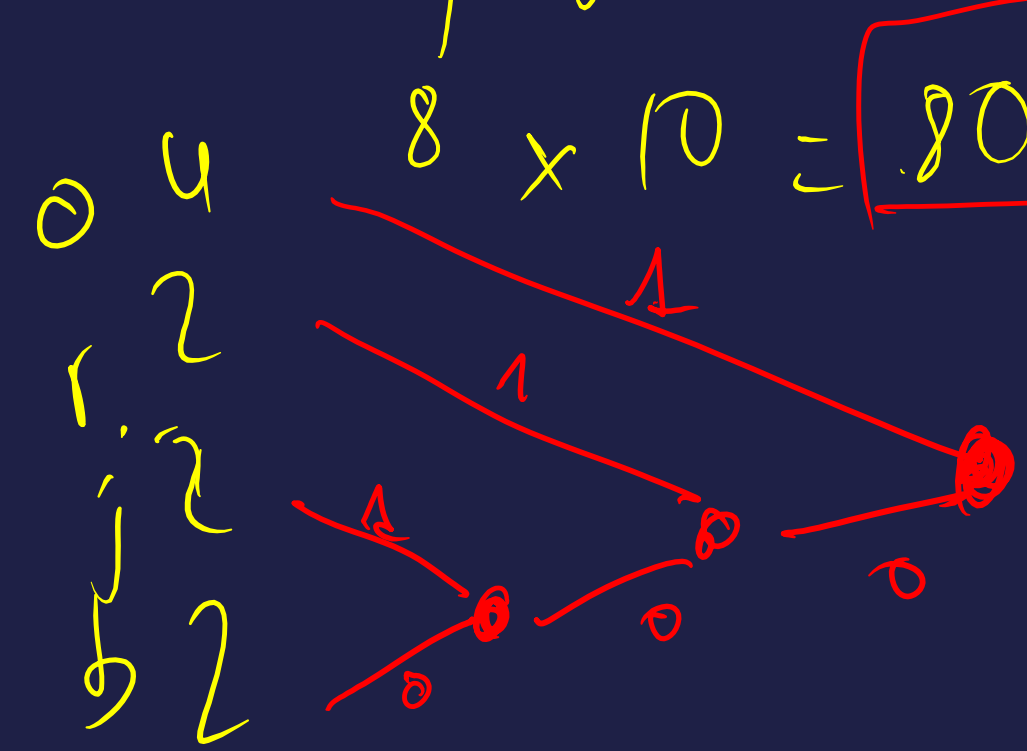
Le code est déterminé à partir d'une estimation des probabilités d'apparition des symboles de source, un code court étant associé aux symboles de source les plus fréquents.

XONATIS

'bojor bojor'

8 x 10 = 80 bits

20 bits

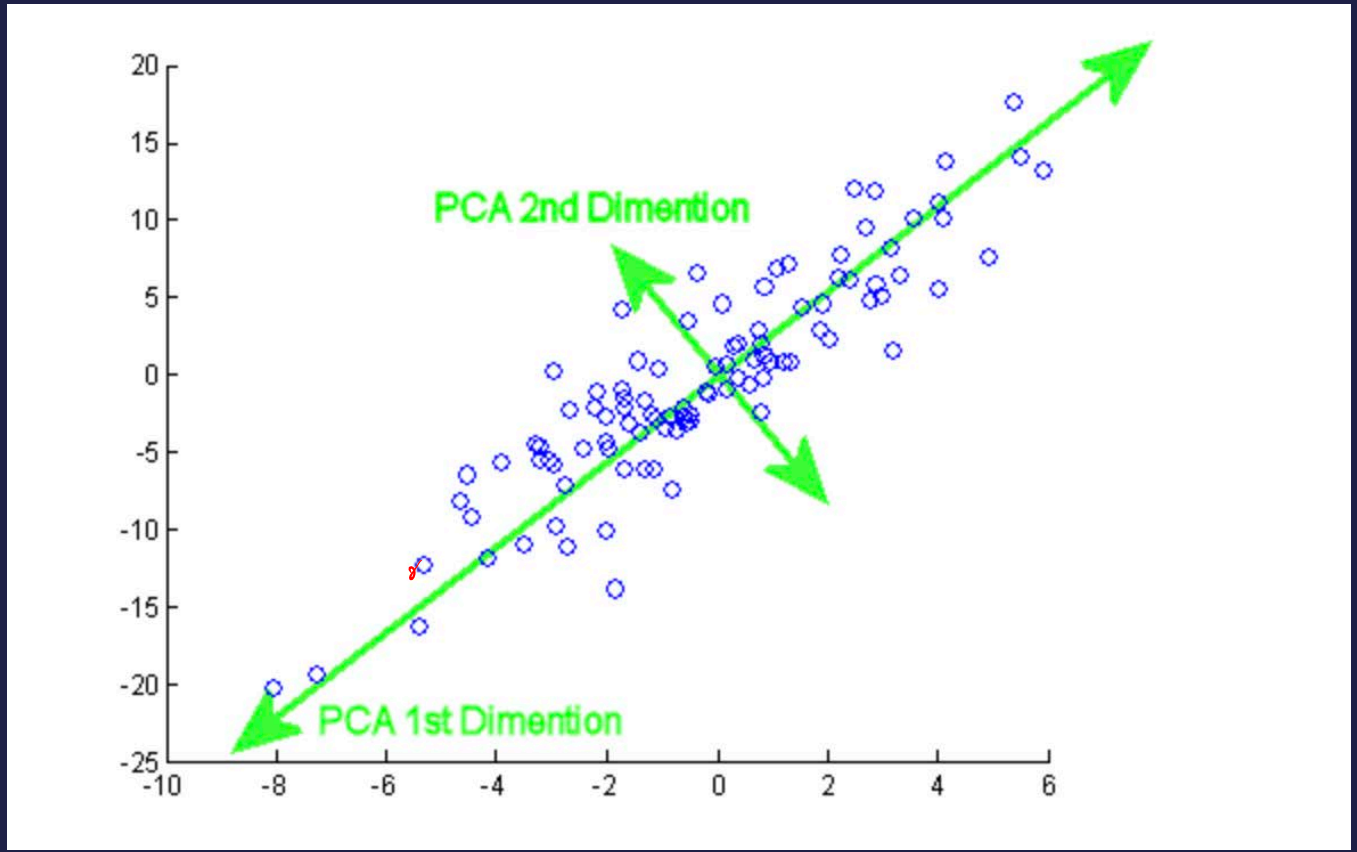


0	1
r	01
j	001
b	000

4	4
2	4
2	6
2	6

Analyse en composantes principales (PCA)

XENATIS


PCA 2nd Dimention
PCA 1st Dimention
20
15
10
5
0
-5
-10
-15
-20
-25
-10
-8
-6
-4
-2
0
2
4
6

XWIKI SAS

XWIKI SAS

XWIKI SAS

XWIKI SAS

XWIKI SAS

XNATIS

La PCA peut être considéré comme un algorithme de compression avec perte, qu'il peut être facilement compréhensible.

La PCA vise a décrire les directions des données représentées dans un espace vectoriel et non plus les données elles-mêmes.

XKNATIS

1. Récursivité
2. Tri
3. DFS
4. Compression
5. Décisionnel

XDNATIS

DECISIONNEL

XWIKI NATIS

Un système expert est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier. Il s'agit de l'une des voies tentant d'aboutir à l'intelligence artificielle.

XWIKI NATIS

Un système expert a pour ambition de résoudre un problème décisionnel, c'est-à-dire dont la réponse est soit vrai soit faux.

XWIKI NATIS

Qu’elle est la formalisation d’un prédicat ?

XWIKI NATIS

Qu’elle est la formalisation d’une proposition ?

XWIKI NATIS

Qu’est-ce qu’un syllogisme ?

XONATIS

XWIKI NATIS

Tous les chiens sont des mammifères
Il n’existe pas de mammifères qui soient des oiseaux

Quelle conclusion ?

XNATIS

Raisonnement qui se formalise par :

$[\forall x\,(estChien(x) \Rightarrow estMammifere(x)) \wedge \neg\exists y\,(estMammifere(y) \wedge estOiseau(y))] \rightarrow \neg\exists z\,(estChien(z) \wedge estOiseau(z))$ qui se lit

("pour tout x, si x est un chien alors x est un mammifère" et "il est faux qu'il existe y qui est à la fois un mammifère et un oiseau") implique qu'il est faux qu'il existe z qui est à la fois un chien et un oiseau".

XWIKI NATIS

Un système part d'une prémisse et combiné avec des règles, déduit une conclusion qui est elle-même une nouvelle prémisse.

Ces systèmes peuvent s'appuyer sur la logique d'ordre 0 (propositions) ou la logique d'ordre 1 (prédicats).