

# Projet – L3 – S2

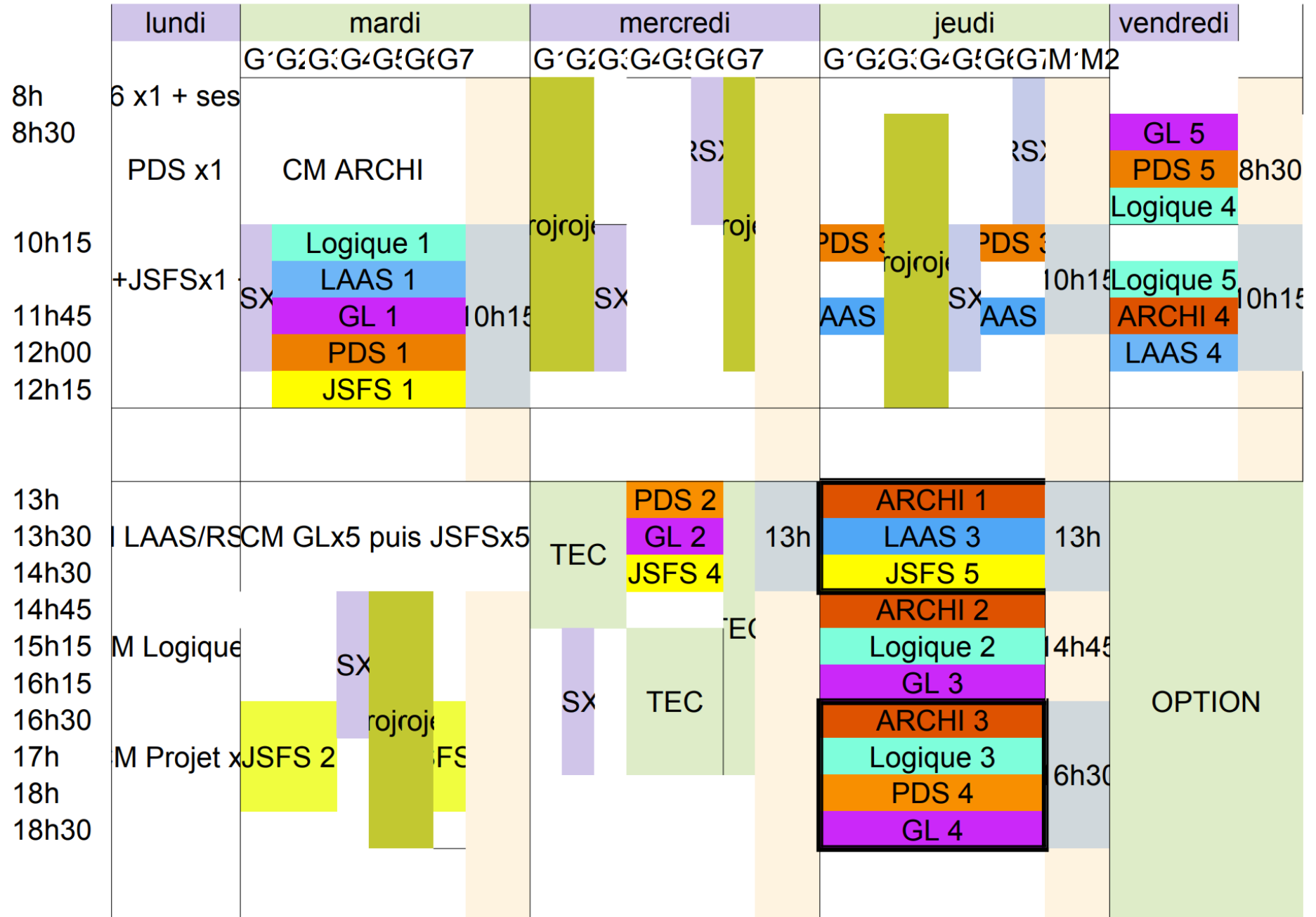
2021-2022



# Organisation

- Les cours auront lieu les 3, 10 et 17 janvier 2021.
- TP (42 h TP/groupe)
  - A partir de la semaine du 10 janvier 2021 (mardi après-midi et mercredi matin et jeudi matin)
  - 2 groupes le mardi 14h45-18h30 (Chaabane (G5), Pierre Tirilly (G6))
  - 3 groupes le mercredi 8h-11h45 (Chaabane (G1), Xavier (G7), Maxime (G2))
  - 2 groupes le jeudi 8h-11h45 (Jean-Claude (G4), Pierre-Allegraud(G3))
  - 11 séances de 3,5 heures
- Cours (4,5 h) :
  - 3 séance de 1,5 heures
  - 03/01/2021 : 08h30 - 10h (SN1 – Amphi Buffon)
  - 10/01/2021 : 16h30 – 18h (SN1 – Amphi Buffon)
  - 17/01/2021 : 16h30 – 18h (M1 – Amphi Galois)

## S6 INFO



	dim. 2/1	lun. 3/1	mar. 4/1	mer. 5/1	jeu. 6/1	ven. 7/1	
08:00				08:00 – 11:45 TP Projet - M5 A11		08:30 – 10:00 TP GL (5) -	
09:00		08:30 – 10:00 Cours JSFS - SN1 Amphi BUFFON	08:30 – 10:00 Cours Projet - SN1 Amphi BUFFON				
10:00		10:15 – 11:45 Cours PDS - SN1 Amphi BUFFON	10:15 – 11:45 TD PDS (1) - M5 A2		10:15 – 11:45 TDTP LAAS (2) -		
11:00							
12:00							
13:00		13:00 – 14:30 Cours LAAS - SN1 BUFFON	13:00 – 14:30 Cours GL - SN1 Amphi MAIGE			13:00 – 14:30 Cours RSX2 - M1 GALOIS	
14:00							
15:00		14:45 – 16:15 Cours Logique - SN1 BUFFON		14:45 – 16:15 Cours Archi - SN1 BUFFON	14:45 – 16:15 TP GL (3) -	14:45 – 16:15 Cours LAAS - M1 GALOIS	
16:00							
17:00		16:30 – 18:00 Cours GL - SN1 Amphi BUFFON	16:30 – 18:00 TP JSFS (2) - M5 A12		16:30 – 18:00 TDTP PDS (4) -		
18:00							

	dim. 2/1	lun. 3/1	mar. 4/1	mer. 5/1	jeu. 6/1	ven. 7/1
08:00						
09:00		08:30 – 10:00 Cours JSFS - SN1 Amphi BUFFON	08:30 – 10:00 Cours Projet - SN1 Amphi BUFFON			08:30 – 10:00 TP GL (5) -
10:00		10:15 – 11:45 Cours PDS - SN1 Amphi BUFFON	10:15 – 11:05 TP Logique (1) - M5 JSFS		10:15 – 11:45 TDTP PDS (3) -	10:15 – 11:05 TP ARCHI -
11:00			11:05 – 12:00 TP M5 A15			11:05 – 12:00 TDTP LAAS (4) -
12:00						
13:00		13:00 – 14:30 Cours LAAS - SN1 BUFFON	13:00 – 14:30 Cours GL - SN1 Amphi MAIGE	13:00 – 13:00 - 1 TDTP PDS (2) - M5 JSFS	13:00 – 13:00 - 1 TDTP LAAS (3) - ARCHI (1) -	13:00 – 14:30 Cours RSX2 - M1 GALOIS
14:00				13:00 – 14:00 M5 A15		
15:00		14:45 – 16:15 Cours Logique - SN1 BUFFON	14:45 – 18:00 TP Projet -	14:45 – 16:15 Cours Archi - SN1 BUFFON	14:45 – 16:15 TP ARCHI (2) -	14:45 – 16:15 Cours LAAS - M1 GALOIS
16:00						
17:00		16:30 – 18:00 Cours GL - SN1 Amphi BUFFON			16:30 – 17:00 TP GL (4) -	16:30 – 17:00 TP Logique (3) -

Groupe 1

Groupe 2

Groupe 3

Groupe 4

Groupe 5

Groupe 6

Groupe 7

Aujourd'hui



2 – 8 janv. 2022



dim. 2/1

lun. 3/1

mar. 4/1

mer. 5/1

jeu. 6/1

ven. 7/1

08:00

08:30 – 10:00

Cours JSFS -  
SN1 Amphi  
BUFFON

08:30 – 10:00

Cours Projet -  
SN1 Amphi  
BUFFON

08:00 – 11:45

TP Projet -

08:30 – 10:00

TDTP PDS (5) -

09:00

10:00

10:15 – 11:45

Cours PDS -  
SN1 Amphi  
BUFFON

10:15 – 12:15

TP JSFS (1) -  
M5 A15

11:00

12:00

13:00

13:00 – 14:30

Cours LAAS -  
SN1 BUFFON

13:00 – 14:30

Cours GL - SN1  
Amphi MAIGE

13:00 – 14:30

TP GL (2) - M5  
A13

13:00 – 14:30

Cours RSX2 -  
M1 GALOIS

14:00

15:00

14:45 – 16:15

Cours Logique -  
SN1 BUFFON

14:45 – 16:15

Cours Archi -  
SN1 BUFFON

14:45 – 16:15

TP GL (3) -

14:45 – 16:15

Cours LAAS -  
M1 GALOIS

16:00

16:30 – 18:00

Cours GL - SN1  
Amphi BUFFON

16:30 – 18:00

TP Logique (3)

-

17:00

# Structure de données : Arbre-B

- Arbre n-aire équilibré
- Recherche en  $\text{Log}(n)$ .  $n$  : nombre de clés.
- Populaire dans les bases de données
- Inventé par Rudolf Bayer et Edward Meyers McCreight, 1970

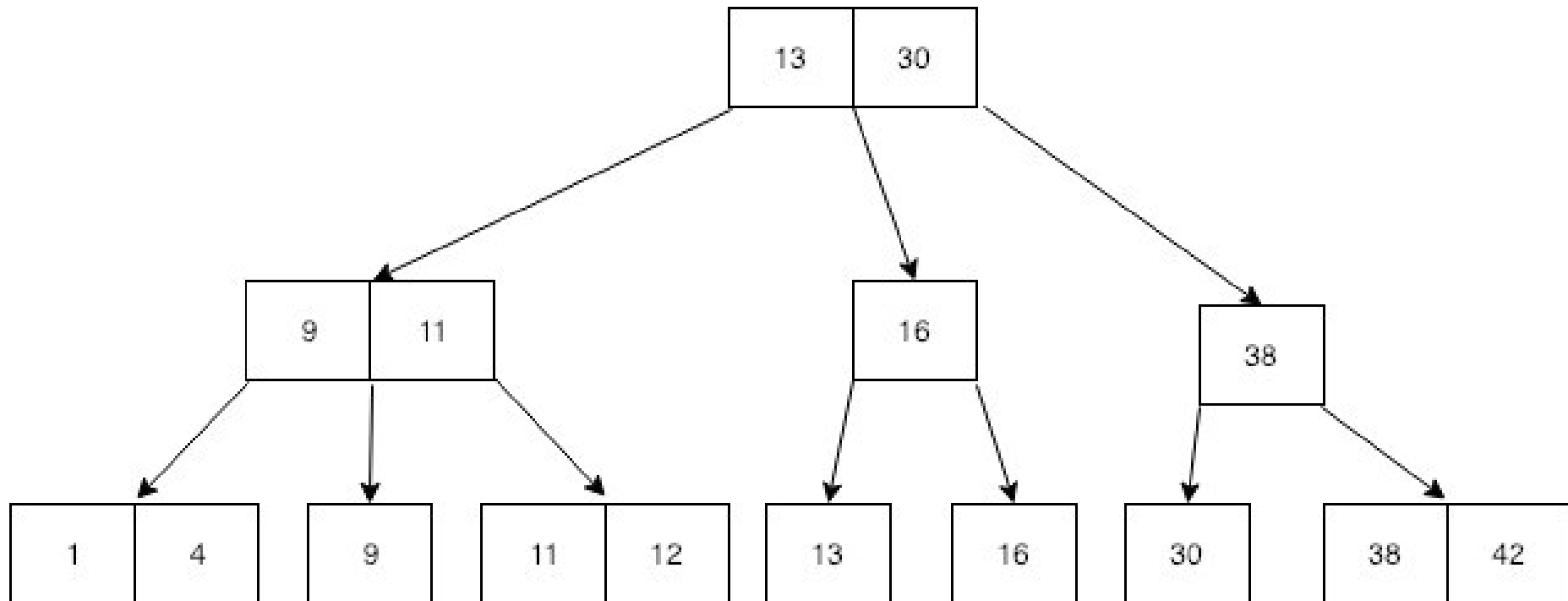
# Structure de données : Arbre-B

- Arbre-B définit par 2 paramètres :  $L$  et  $U$ .  $U$  est le nombre de nœuds fils max et  $L$  le nombre de nœuds fils min. Arbre-B ( $L$ ,  $U$ ).
- $n$  = nombre de clés contenues dans le nœud  $x$
- $n$  clefs notées  $c[1], \dots, c[n]$
- Un booléen indiquant si  $x$  est une feuille ou non
- $n+1$  fils :  $f[1], \dots, f[n+1]$ . Une feuille ne contient pas de fils
- L'arbre-B vérifie les propriétés suivantes :
  - Toutes les feuilles ont la même profondeur, à savoir la hauteur  $h$  de l'arbre
  - Si  $x$  n'est pas une feuille :
    - pour  $2 \leq i \leq n$ , pour toute clef  $x$  du fils  $i$  :  $c[i-1] \leq x \leq c[i]$
    - Pour toute clef  $x$  du fils  $1$  :  $x \leq c[1]$
    - Pour toute clef  $x$  du fils  $n+1$  :  $c[n] \leq x$
  - Si  $x$  n'est pas la racine,  $n$  est compris entre  $L-1$  et  $U-1$ .



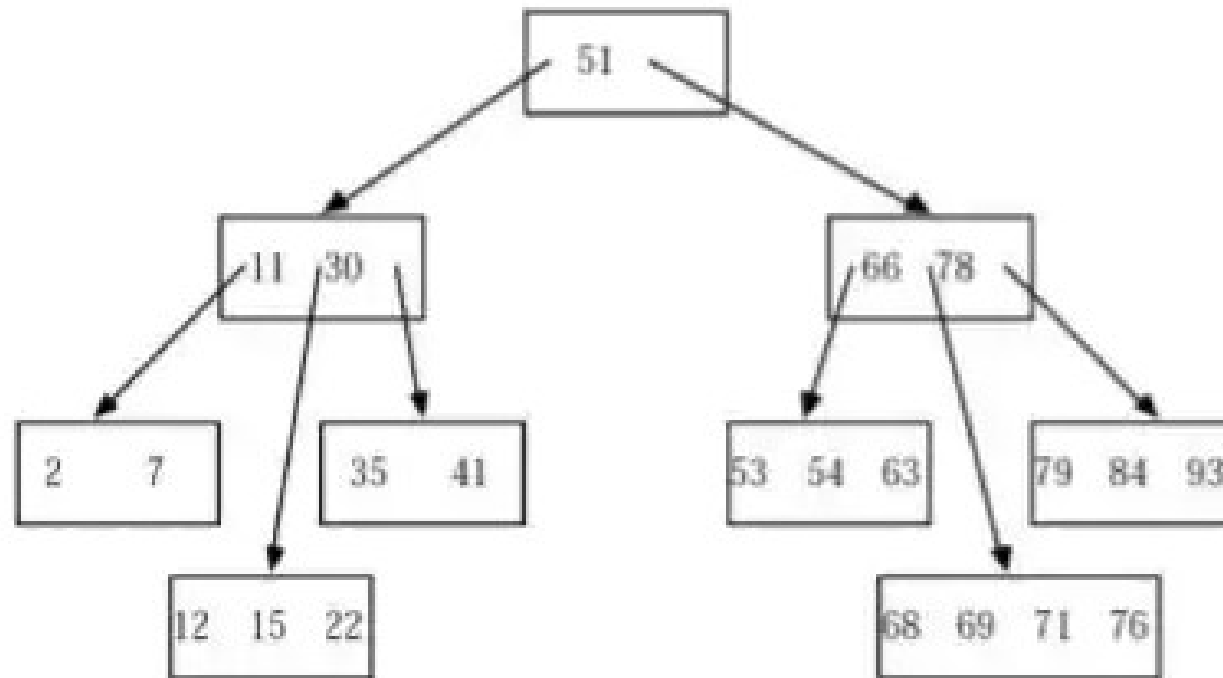
# Structure de données : Arbre-B

- Arbre-B,  $L=2$  et  $U = 3$



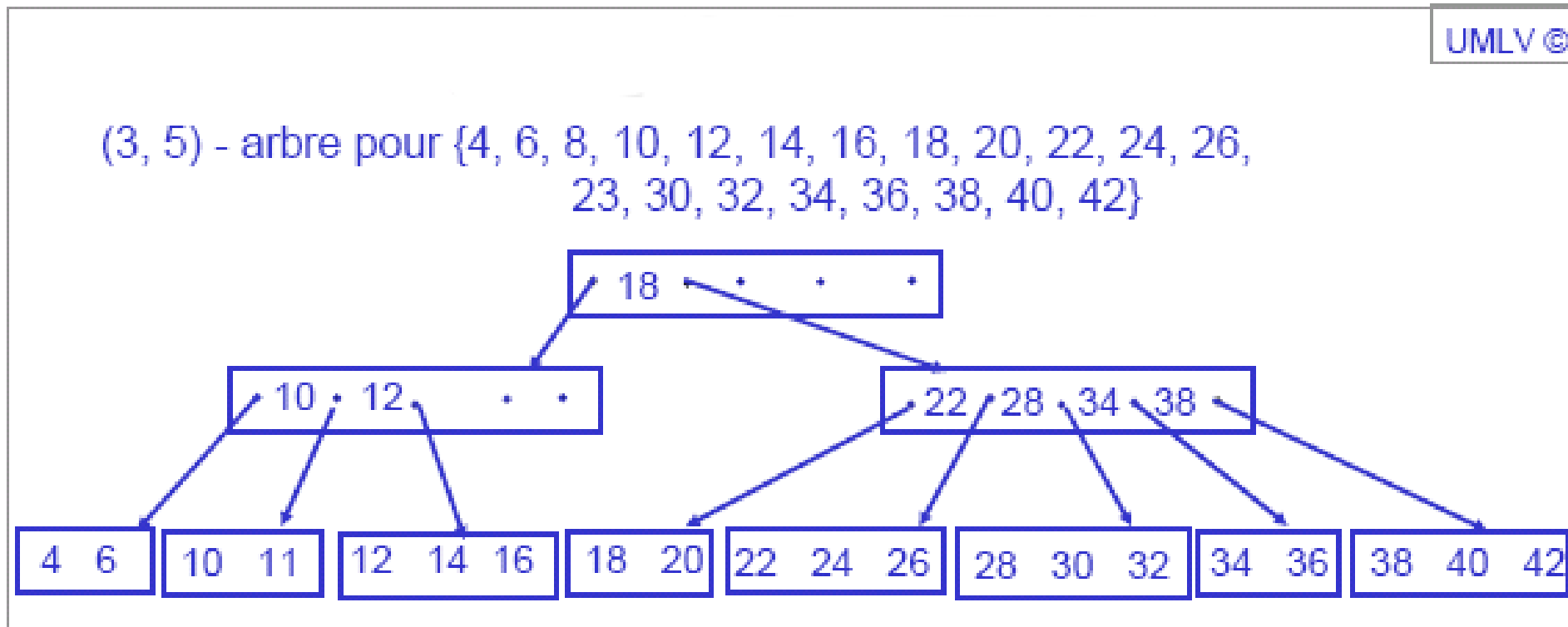
# Structure de données : Arbre-B

- Arbre-B, L=3 et U = 5



# Structure de données : Arbre-B

- Arbre-B,  $L=3$  et  $U = 5$



# Structure de données : Arbre-B

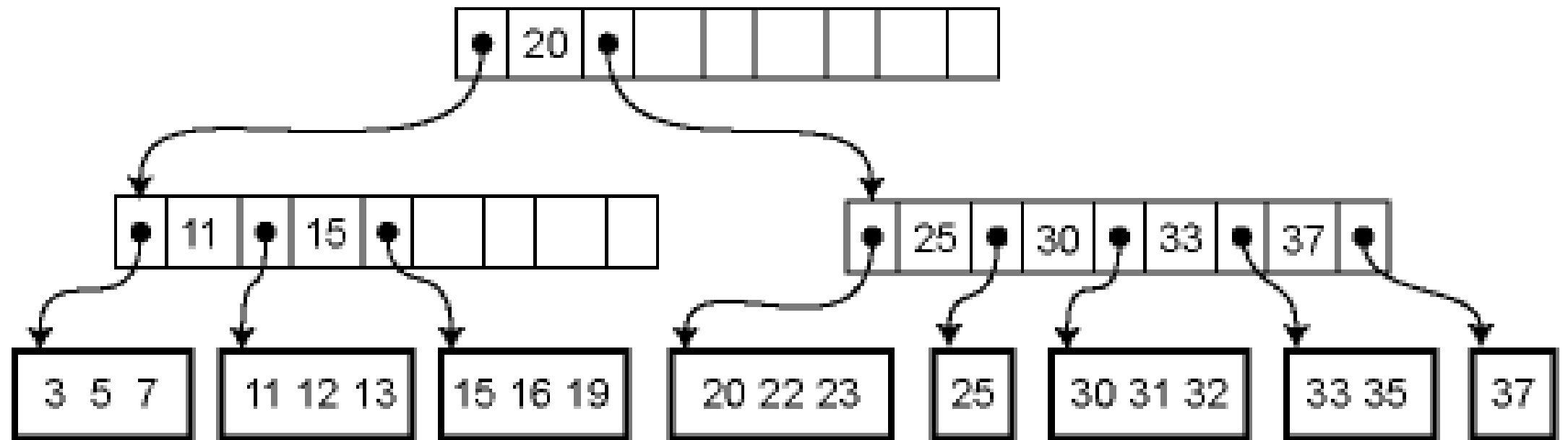
- Structure Nœud {  
    size n; // nombre de clés utilisées  
    bool feuille ; // vrai si le nœud est une feuille  
    Nœud parent; // le parent du nœud, utile pour certains algorithmes  
    int cles [L]; // tableau des clés  
    Nœud fils[]; // tableau des fils, si ce n'est pas une feuille  
    }

# Recherche

- Recherche (nœud-racine, x) : booléen

```
fonction Rechercher(noeud, x):  
    i = 0  
    tant que i < noeud.taille et x > noeud.cles[i]:  
        i += 1  
  
    si noeud.feuille:  
        retourner x == noeud.cles[i]  
    sinon:  
        si x == noeud.cles[i]:  
            retourner Vrai  
        sinon si i == noeud.taille et x > noeud.cles[noeud.taille - 1]:  
            retourner Rechercher(noeud.branches[noeud.taille], x)  
        sinon:  
            retourner Rechercher(noeud.branches[i], x)
```

# Recherche



Insert(13)

# Insertion

- Insertion(nœud-racine, clef) : booleen

# Suppression

- suppression (nœud-racine, clef) : booléen



# Grille d'évaluation

- Livrable = Démonstration + Documentation, à la fin de chaque séance aux cours des deux dernières séances du projet.
- La démonstration est réalisée sur le poste de travail, sur une durée de 30 minutes environ, sur les deux batteries de tests présentées.
- La documentation ne doit pas dépasser 10 pages. Elle doit décrire : l'historique de l'arbre B et les algorithmes (codes Python) commentés : l'algorithme de recherche, l'algorithme d'insertion, l'algorithme de suppression. Chaque algorithme doit comporter la documentation (docstring) : entrée, sortie, exemples, estimation de la complexité temporelle des différents algorithmes.

# Test - 1

- $(L,U)=(2,3)$ . Le noeud contient 2 clés au maximum et 1 clé au minimum.
- Insertion dans l'ordre : 2, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 7, 9, 11, 13.
- Suppression dans l'ordre : 14, 10, 20, 18, 16, 24, 6
- La vidéo ci-dessous visualise l'état de l'arbre à chaque insertions ou suppression.
- <https://www.youtube.com/watch?v=coRJrcIYbF4>

# Test - 2

- Suite à cette première batterie de tests, il convient de réaliser d'autres tests.
- $(L, U) = (6, 11)$ . Le noeud contient 10 clé au maximum et 5 clés au minimum.
- Soit la liste des valeurs suivantes : 10, 20, 30, ... 5000, 5, 15, 25, 35, ..4995.
- Suppression dans l'ordre : 10, 20, 30, ... 5000, 5, 15, 25, 35, ..4995
- Généralisez les tests en considérant  $N = (2, 10, 100, 1000, 10000)$ .

# Règles de programmation

- Les travail réalisé doit être téléversé sur Gitlab, à la fin de chaque séance.
- Le projet peut être réalisé par une personne ou un binôme.
- Il est possible pour l'étudiant de choisir un projet différent. Le sujet en question doit être proposé et validé par l'enseignant responsable du groupe et l'enseignant responsable de l'UE, avant le 30 janvier 2022.
- Avant de se lancer dans la réalisation et le développement des algorithmes du projet, il est nécessaire de prendre le temps de le comprendre et de le dérouler sur des données tests. Ensuite, organiser développer l'architecture en technologie objet du projet (classes et liens entre les classes).