



# Présentation PIM : PageRank

Groupe EF-03

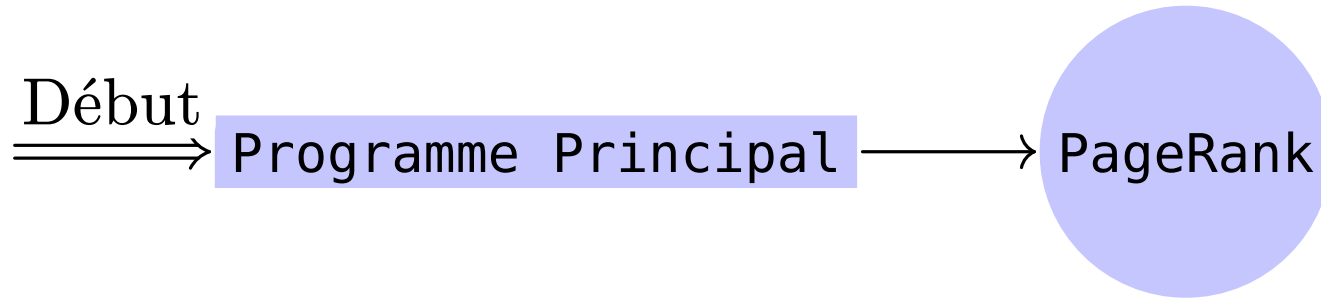
Louis Thevenet

Albin Morisseau

ENSEEIH

17 Janvier, 2024

# Programme principal



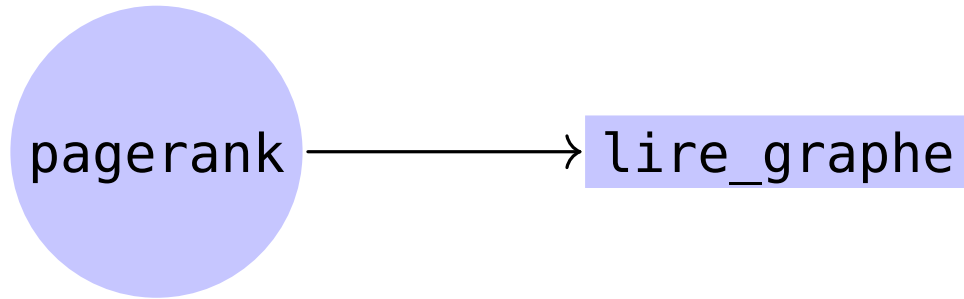
- Programme Principal :
  1. Traitement de la ligne de commande
  2. Appel à PageRank



PageRank

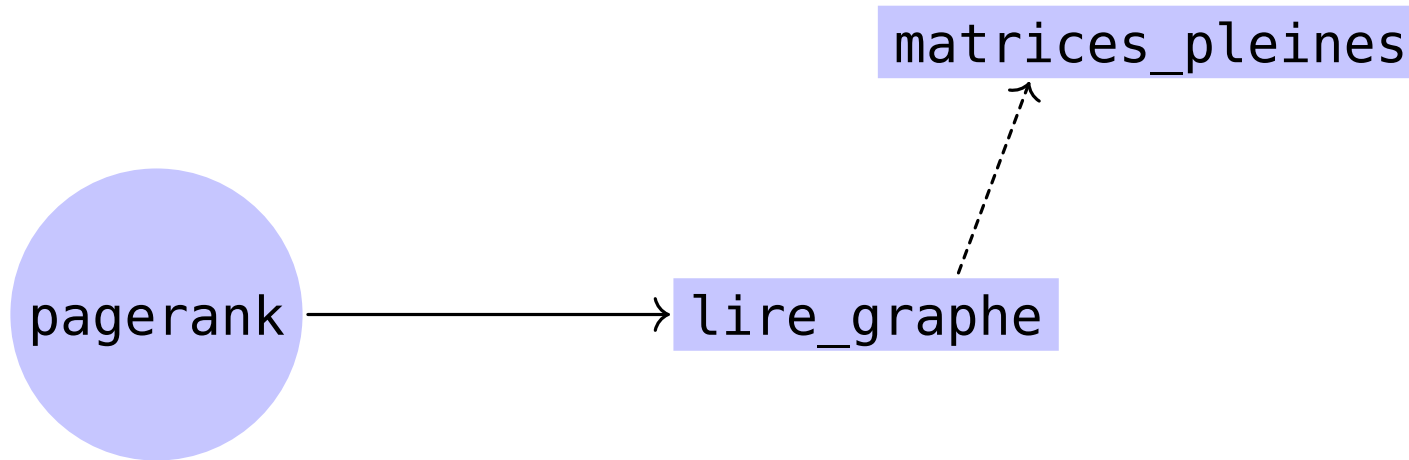
- PageRank :
  1. Traite les données d'entrée
  2. Applique l'algorithme PageRank

# Lire Graphe



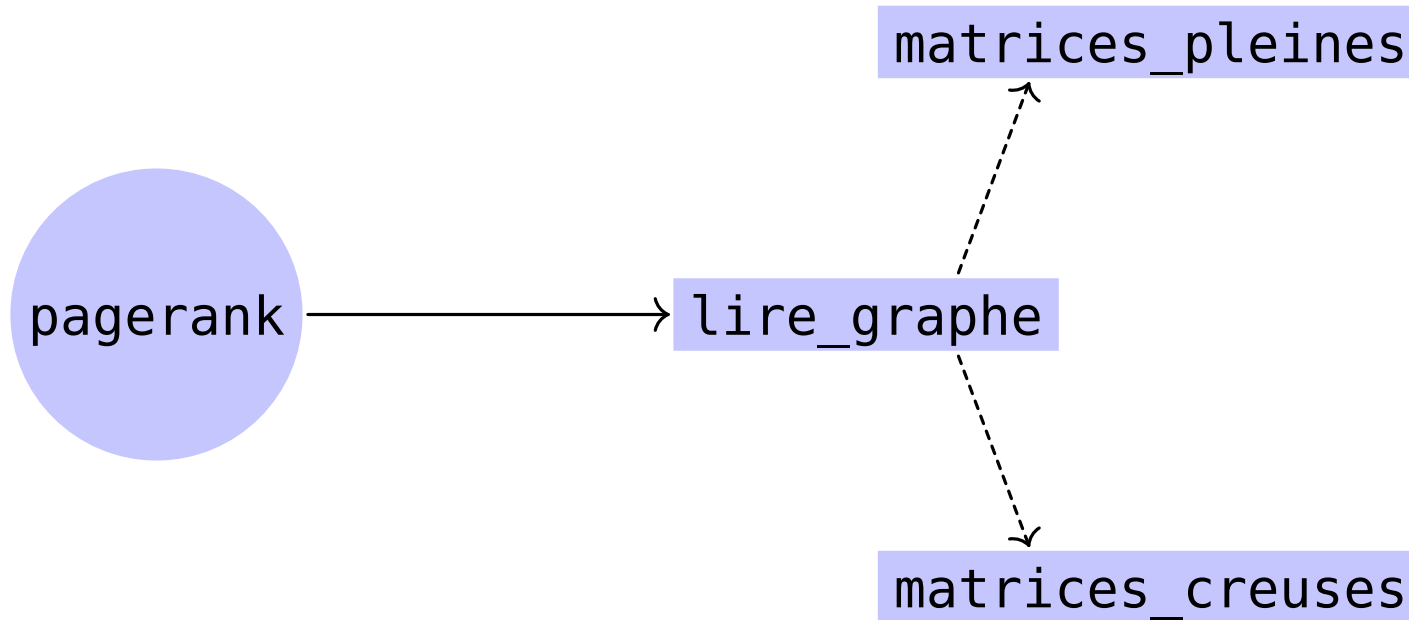
- Traite les données

# Matrices Pleines



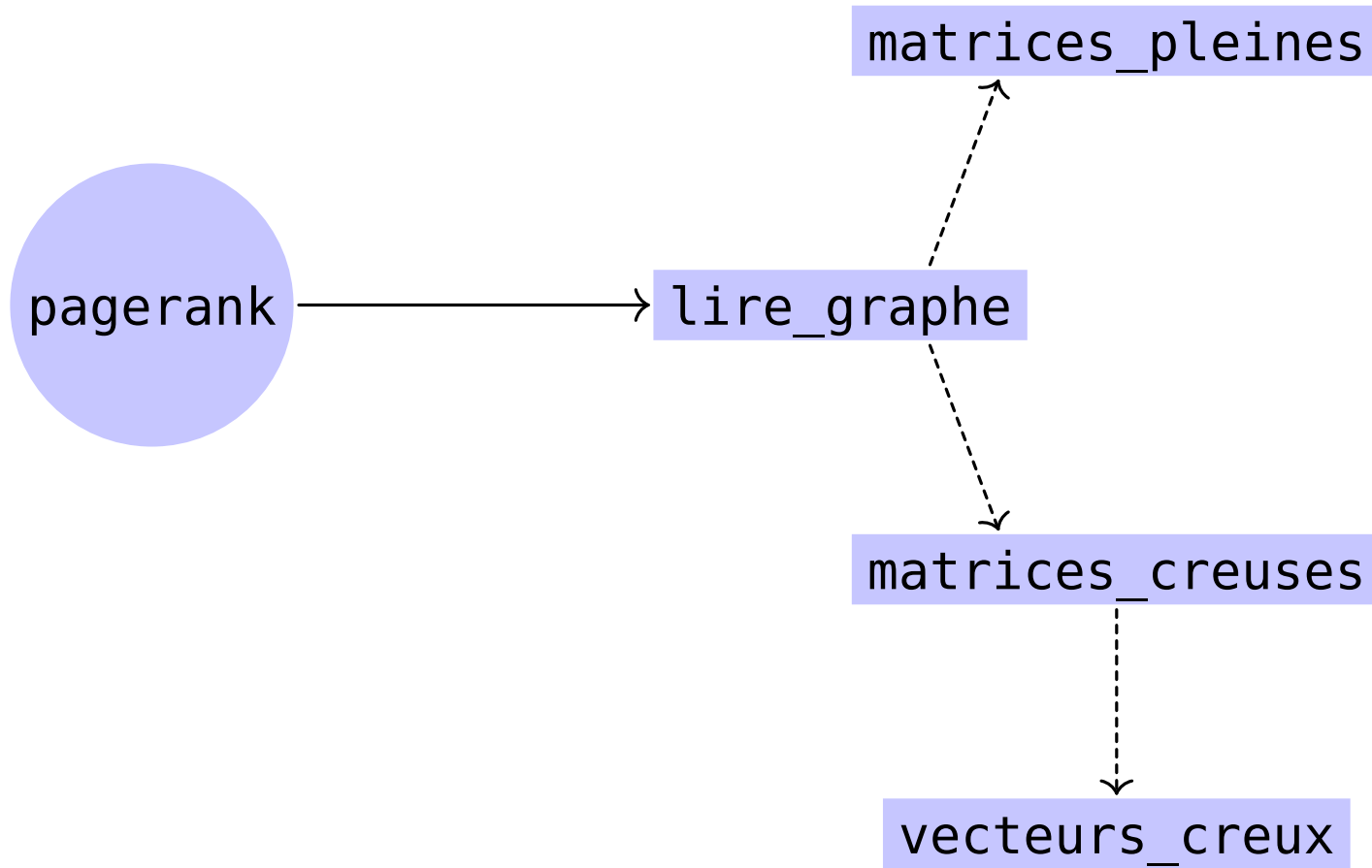
- Traite les données
  - Cas Matrices Pleines
    - Crée la matrice  $G$

# Matrices creuses



- Traite les données
  - Cas Matrices Creuses
    - Crée la matrice  $M$  telle que  $\forall (i, j) \in \llbracket 1, n \rrbracket^2, M_{i,j} \in \{0, 1\}$
    - Crée la vecteur  $F$  tel que  $\forall i \in \llbracket 1, n \rrbracket, F_i = \sum_{k=1}^n M_{i,k}$

# Vecteurs Creux



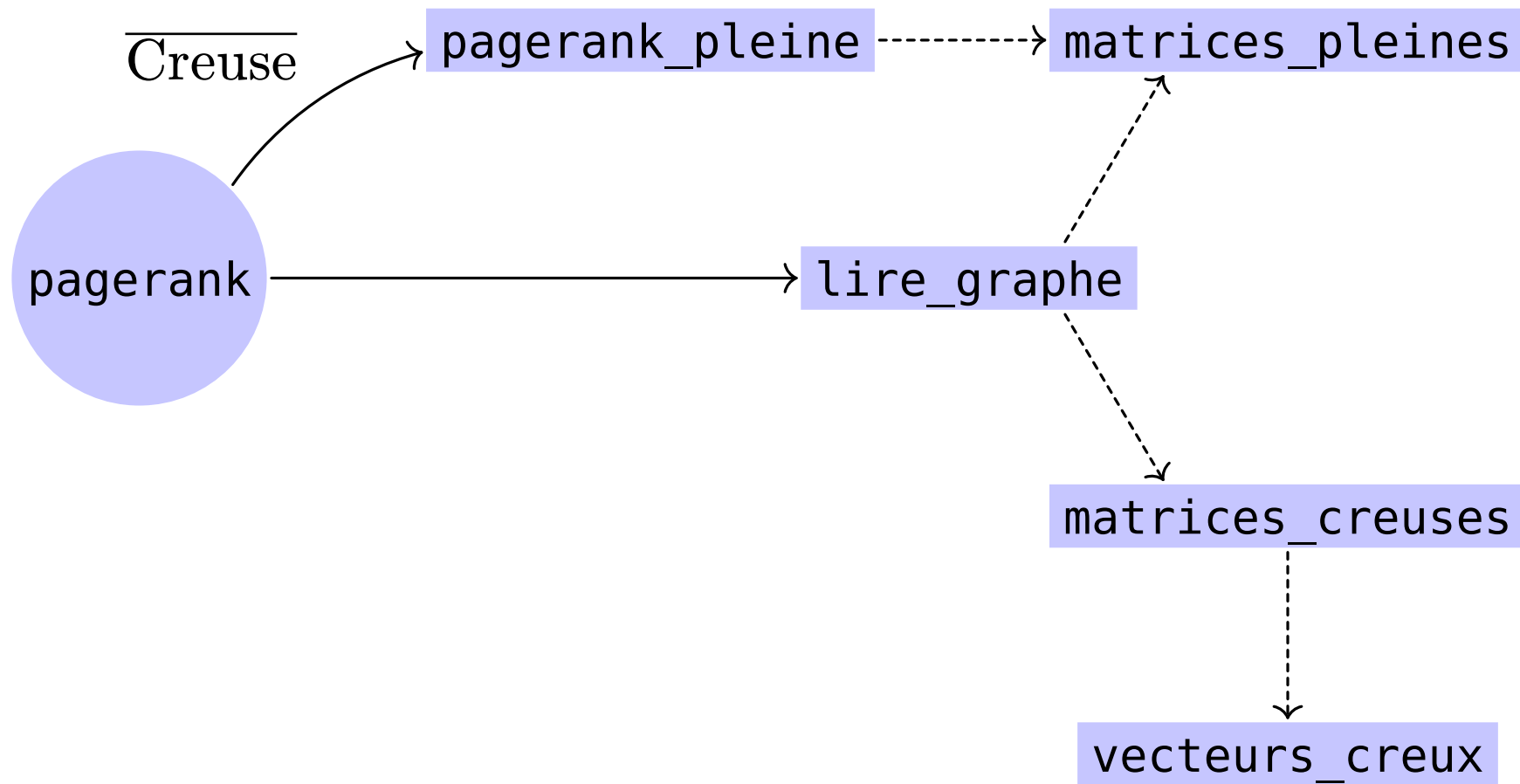
- type `T_Matrice_Creuse` est tableau de `T_Vecteur_Creux`

# Vecteurs Creux : détails

```
1 type T_Cellule;  
2 type T_Vecteur_Creux is access T_Cellule;  
3 type T_Cellule is  
4     record  
5         Indice : Integer;  
6         Valeur : Long_Float;  
7         Suivant : T_Vecteur_Creux;  
8     end record;
```



# PageRank Pleine



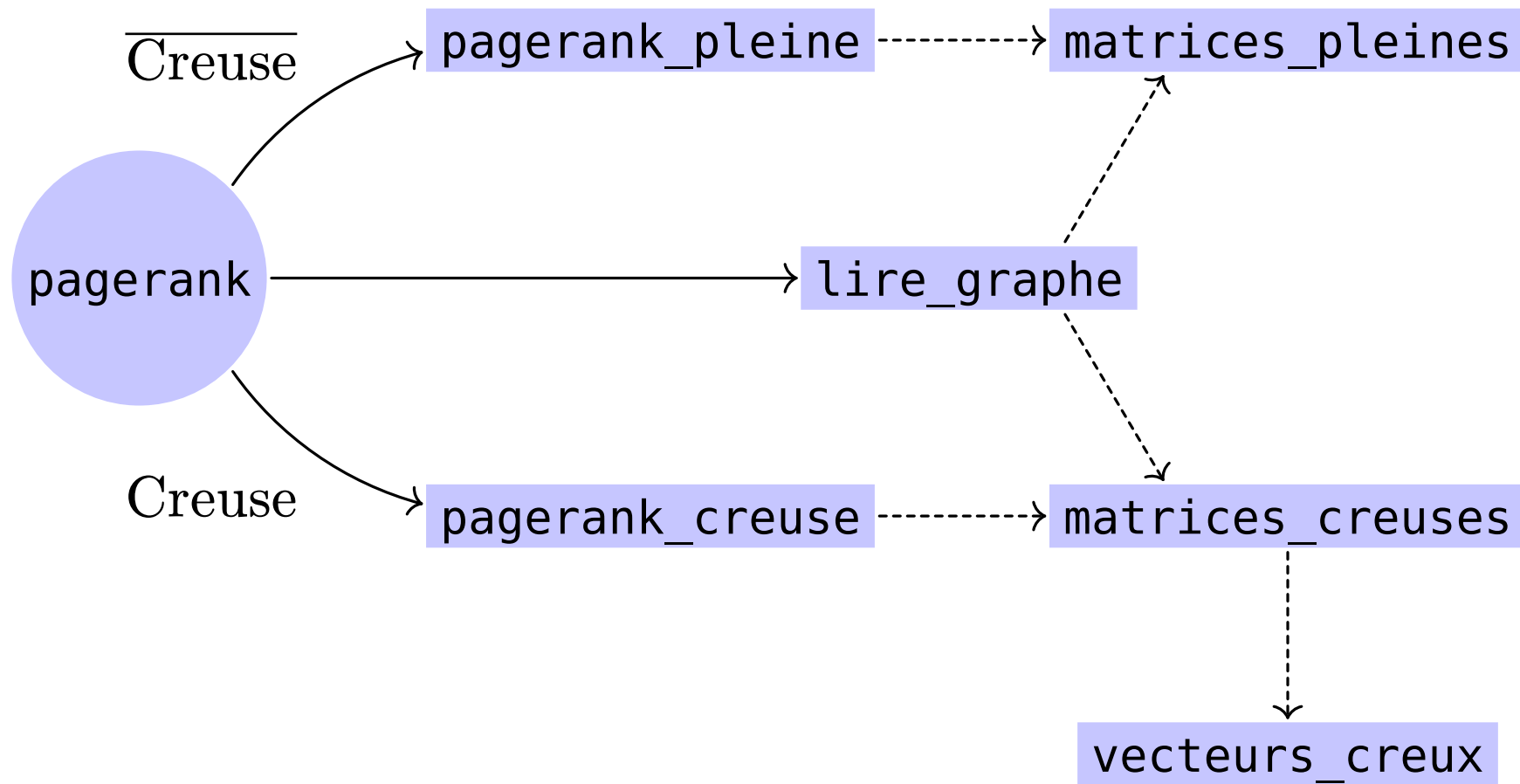
- Réalise l'algorithme PageRank sur une matrice pleine

# PageRank Pleine : détails

Itération de l'algorithme : produit vectoriel de  $\pi_k$  (Poids) et  $G$ .

```
1 for J in 1 .. N loop
2   Resultat(J) := 0.0;
3   for I in 1..N loop
4     Resultat(J) := Resultat(J)+Poids(I)*G(I, J);
5   end loop;
6 end loop;
7 return Resultat;
```

# PageRank Creuse



# PageRank Creuse : détails

Dans le cas Matrices Creuse, on parcourt les lignes avec un curseur.

```
1 for J in 1..Taille loop -- pour chaque colonne
2   Resultat(J) := 0.0;
3   Tete := S(J);
4   for I in 1..Taille loop -- parcourir la ligne
5     if Tete = Null then
6       Tmp := 0.0;
7     elsif Tete.Indice = I then
8       Tmp := Tete.Valeur/Facteurs(I);
9       Tete := Tete.Suivant;
10    else
```

# PageRank Creuse : détails

```
11         while Tete /= Null and then Tete.Indice < I
loop
12             Tete := Tete.Suivant;
13         end loop;
14         Tmp := 0.0;
15     end if;
16     Resultat(J) := Resultat(J) + (Alpha * Tmp+ beta)
    * Poids(I);
17     end loop;
18 end loop;
```

On recrée la matrice  $G$  directement dans les itérations dans la ligne 16.

# Idées d'amélioration

- Données utiles : présence d'au moins un référencement et  $F$  lui-même
- Dans le cas Matrices Creuses, on peut remplacer  $M$  par  $\tilde{M}$  :

$$\forall (i, j) \in \llbracket 1, n \rrbracket^2, \tilde{M}_{i,j} \in \{\text{Vrai}, \text{Faux}\}$$

- Motivations
  - Long\_Float : 64 bits
  - Booléen : 1 bit