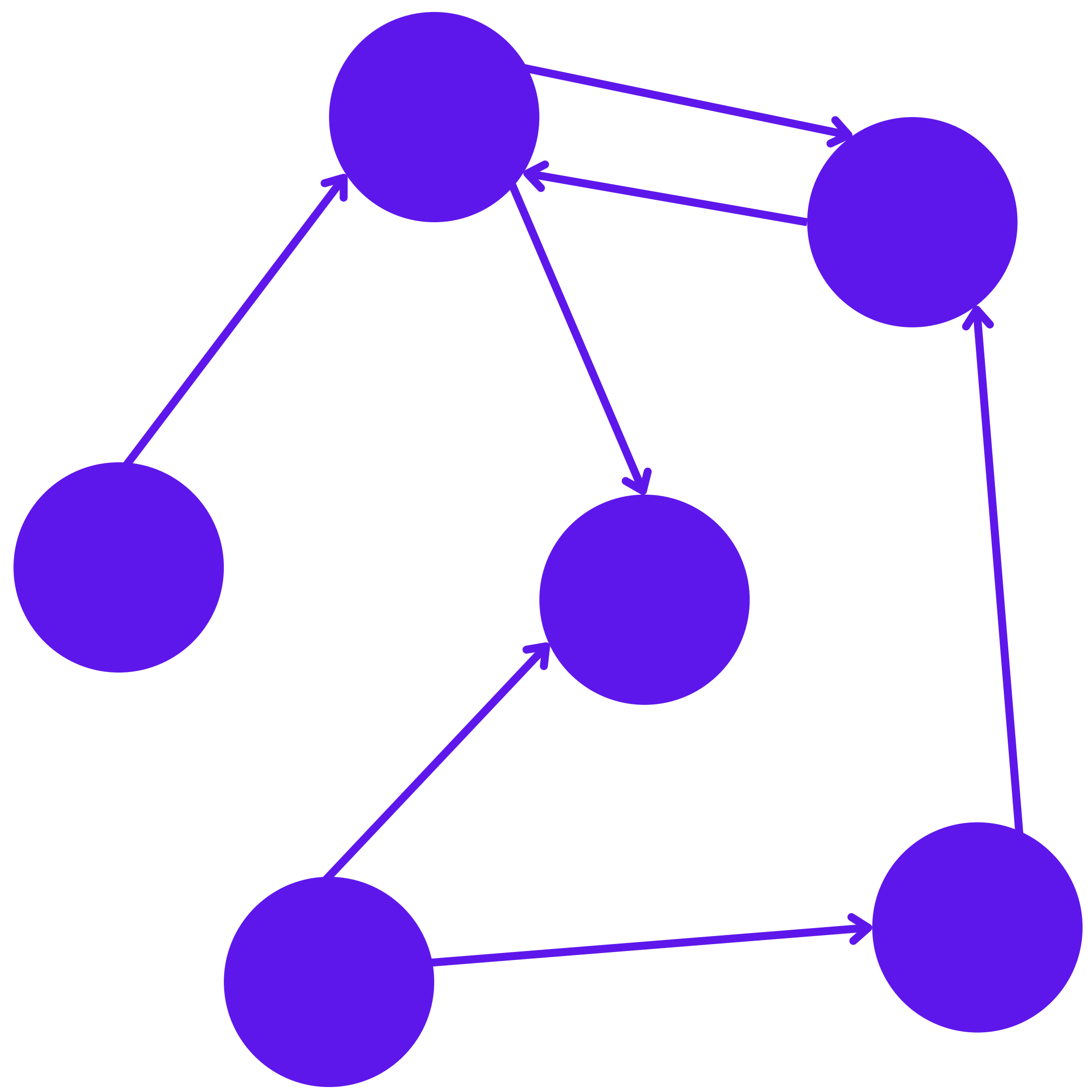
Algorithme de Floyd-Warshall



Projet SM501 - Théorie des Graphes

MEMBRES

Eya LAHOUEL
Myriem TIH
Zoubir SAHNOUN
Mohamed Hedi MOURALI
Mohamed Ilyes BEN SAID
Mohamed SAKHO

Objectif

Objectif 1 : Implémentation de l'algorithme de Floyd-Warshall

- Calculer les plus courts chemins entre tous les couples de sommets
- Traiter des graphes orientés et valués

Objectif 2 : Développer une interface de visualisation

- Charger les graphes depuis des fichiers
- Afficher les matrices de distances et de chemins
- Visualiser les résultats de manière claire

Objectif 3 : Application concrète

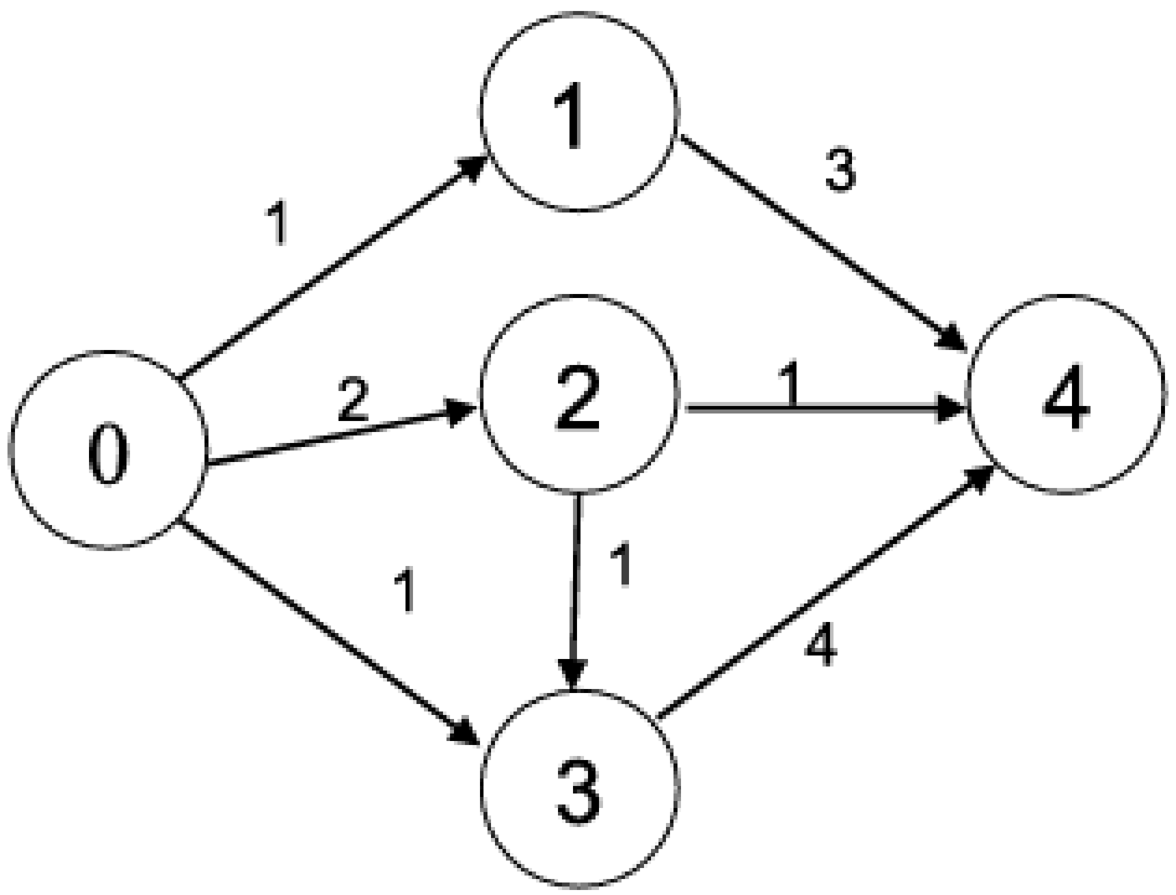
- Démontrer l'utilité pratique de l'algorithme
- Cas d'usage : modélisation de la propagation virale

Représentation des Graphes

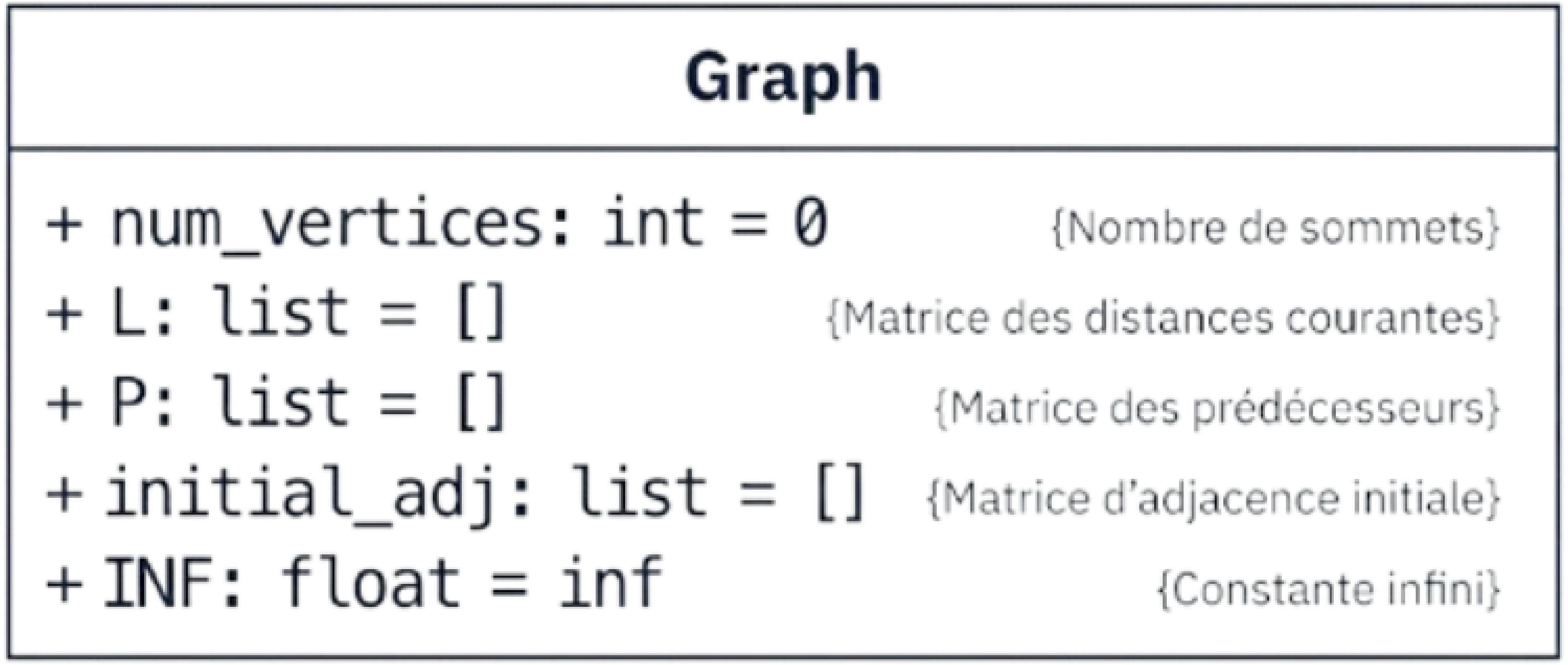
Structure :

1. Nombre de sommets (N) ⟶ 5
2. Nombre d'arcs (M) ⟶ 7
3. Liste des arcs : ⟶
   `Source Dest Poids`

```
0 1 1
0 2 2
0 3 1
1 4 3
2 3 1
2 4 1
3 4 4
```



Representation d’un graphe en mémoire



Chargement (load_from_file)

- Lecture du fichier texte
- Initialisation des structures de données
- Stockage du graphe en mémoire

Traitement (floyd_warshall)

- Exécution de l'algorithme
- Calcul des plus courts chemins

Reconstruction (get_all_shortest_paths)

- Reconstruction des chemins optimaux
- Affichage des résultats

L’algorithme de Floyd warshall

```
{init}
POUR i ← 1 à n FAIRE
        POUR j ← 1 à n FAIRE
                {
                L[i,j] ← coût(i,j,G)
                P[i,j] ←i
                }


{calcul des plus courts chemins}
POUR k ← 1 à n FAIRE
        POUR i ← 1 à n FAIRE
                POUR j ← 1 à n FAIRE
                        SI L[i,k] + L[k,j] < L[i,j] ALORS
                                L[i,j] ← L[i,k] + L[k,j]
                                P[i,j] ← P[k,j]
                        fin
                fin
        fin
```

Triple boucle imbriquée

Complexité $O(N^3)$

Matrices L & P

Matrice L

	0	1	2	3	4
0	0	1	2	1	∞
1	∞	0	∞	∞	3
2	∞	∞	0	1	1
3	∞	∞	∞	0	4
4	∞	∞	∞	∞	0

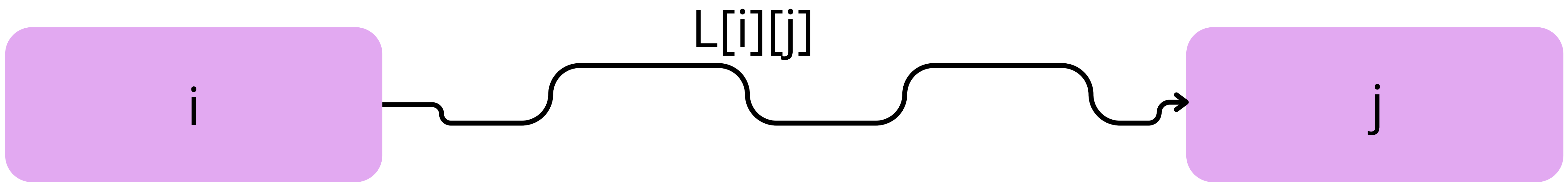
Matrice des coûts

Matrice P

	0	1	2	3	4
0	ø	0	0	0	ø
1	ø	ø	ø	ø	1
2	ø	ø	ø	2	2
3	ø	ø	ø	ø	3
4	ø	ø	ø	ø	ø

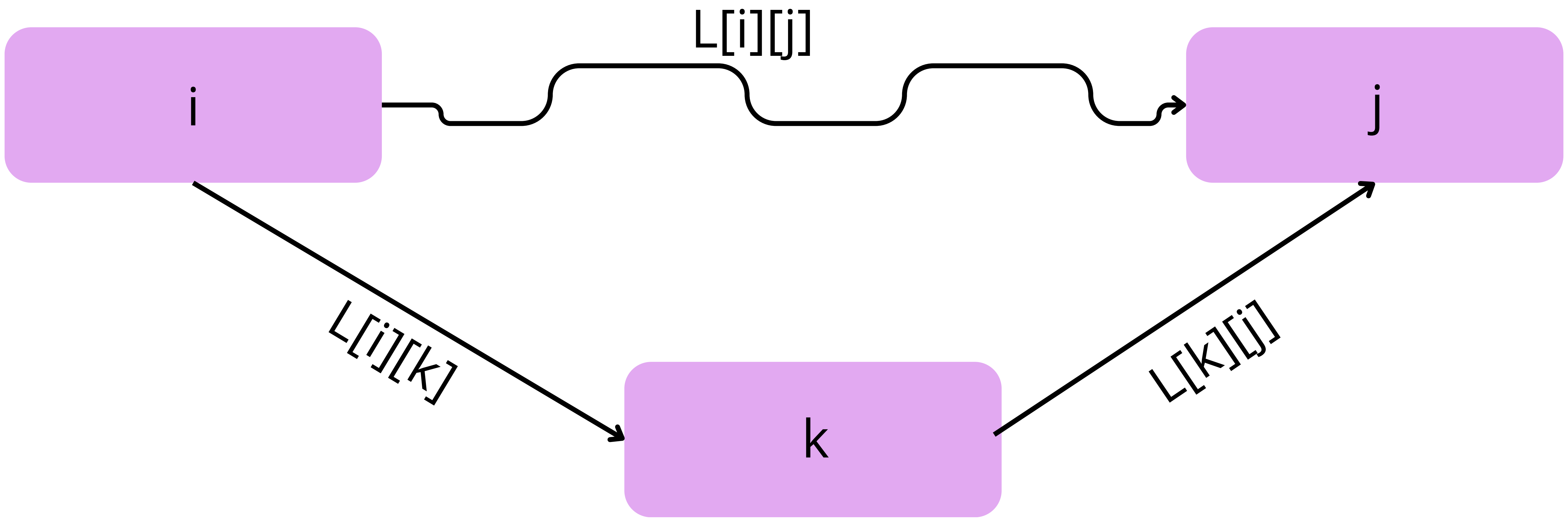
Matrice des predecesseurs

L’algorithme de Floyd warshall



L’algorithme de Floyd warshall

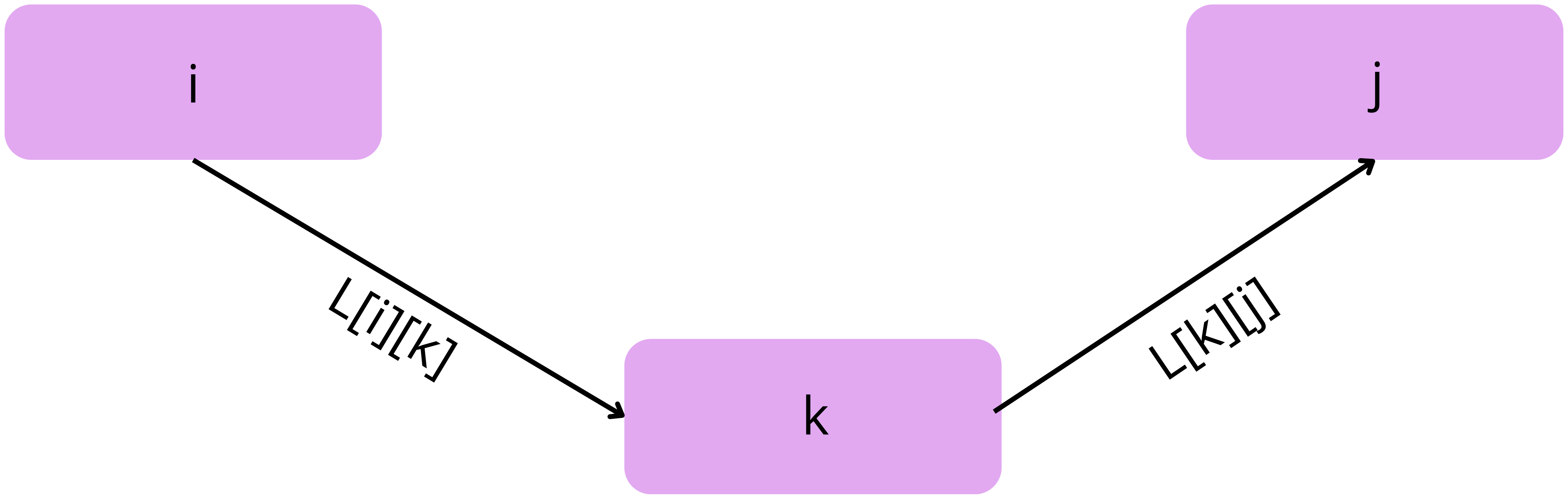
Si L[i][k] + L[k][j] < L[i][j]:



L’algorithme de Floyd warshall

Si L[i][k] + L[k][j] < L[i][j]:

L[i][j] = L[i][k] + L[k][j]

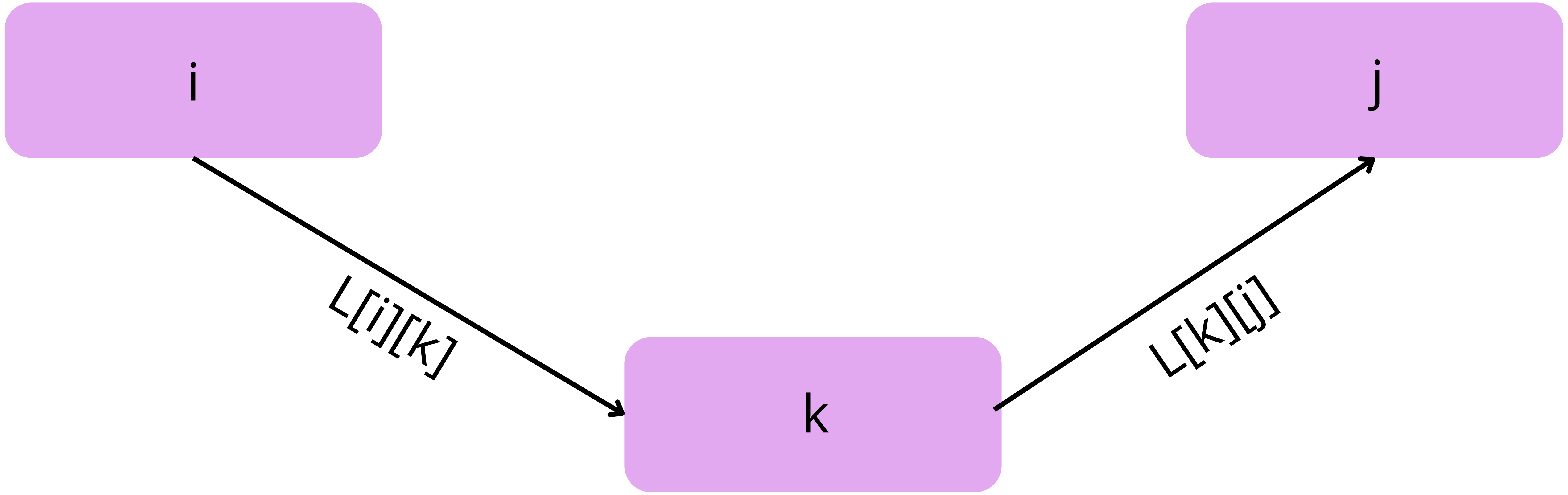


L'algorithme de Floyd warshall

Si L[i][k] + L[k][j] < L[i][j]:

L[i][j] = L[i][k] + L[k][j]

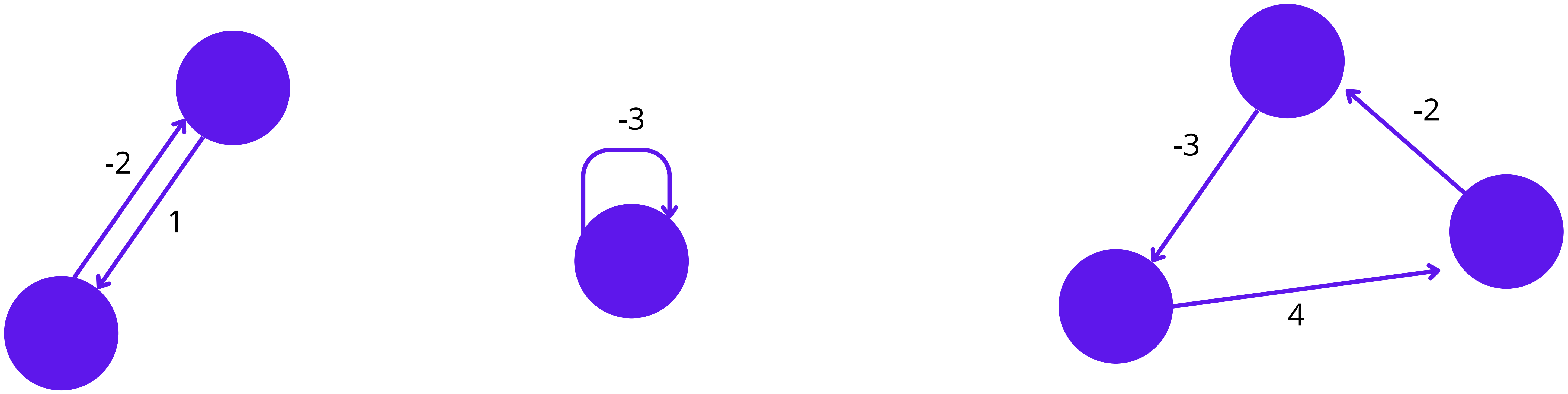
P[i][j] = P[k][j]



Detection des cycles absorbants

Cycle absorbant = Cycle dont la somme des poids est < 0

Ils permettent de réduire indéfiniment le coût d’un trajet et donc pas de plus court chemin



Detection des cycles absorbants

Lorsqu'un sommet fait parti d'un cycle négatif, dist[k][k] devient négatif après execution complète de l'algorithme

On detecte donc les graphes à circuit absorbant on vérifiant la diagonale de la matrice L

Exemple d'une matrice L d'un graphe à circuit absorbant:

	0	**1**	**2**	**3**
0	0	-6	-9	-8
1	∞	-4	-7	-6
2	∞	-5	-8	-7
3	∞	∞	∞	0

Detection des cycles absorbants

Lorsqu’un sommet fait parti d’un cycle négatif, dist[k][k] devient négatif après execution complète de l’algorithme

On detecte donc les graphes à circuit absorbant on vérifiant la diagonale de la matrice L

Exemple d’une matrice L d’un graphe à circuit absorbant:

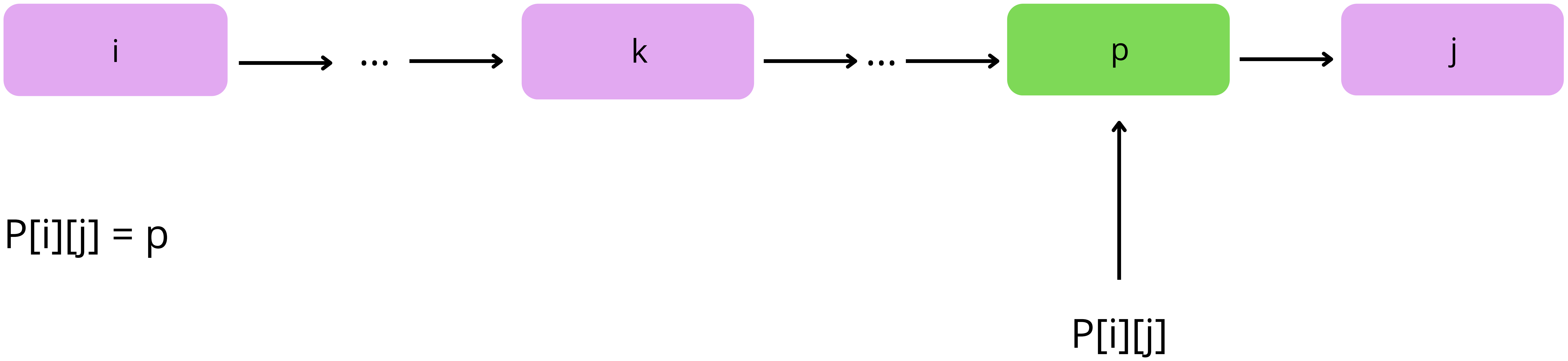
	0	**1**	**2**	**3**
0	0	-6	-9	-8
1	∞	-4	-7	-6
2	∞	-5	-8	-7
3	∞	∞	∞	0

Reconstruction du chemin i → j

$$i \longrightarrow \dots \longrightarrow k \longrightarrow \dots \longrightarrow p \longrightarrow j$$

j]

Reconstruction du chemin i → j



P[i][j] = p

p → j]

Reconstruction du chemin i → j

i → … → k → … → p → j

P[i][j] = p

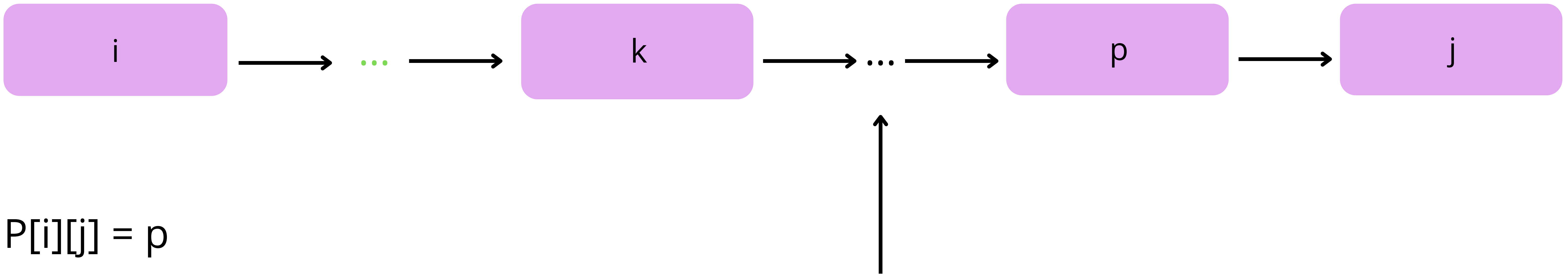
P[i][p] = …

↑

P[i][p]

… → p → j]

Reconstruction du chemin i → j



P[i][...]

P[i][j] = p

P[i][p] = ...

⋮

P[i][...] = i

[i → ... → k → ... → p → j]

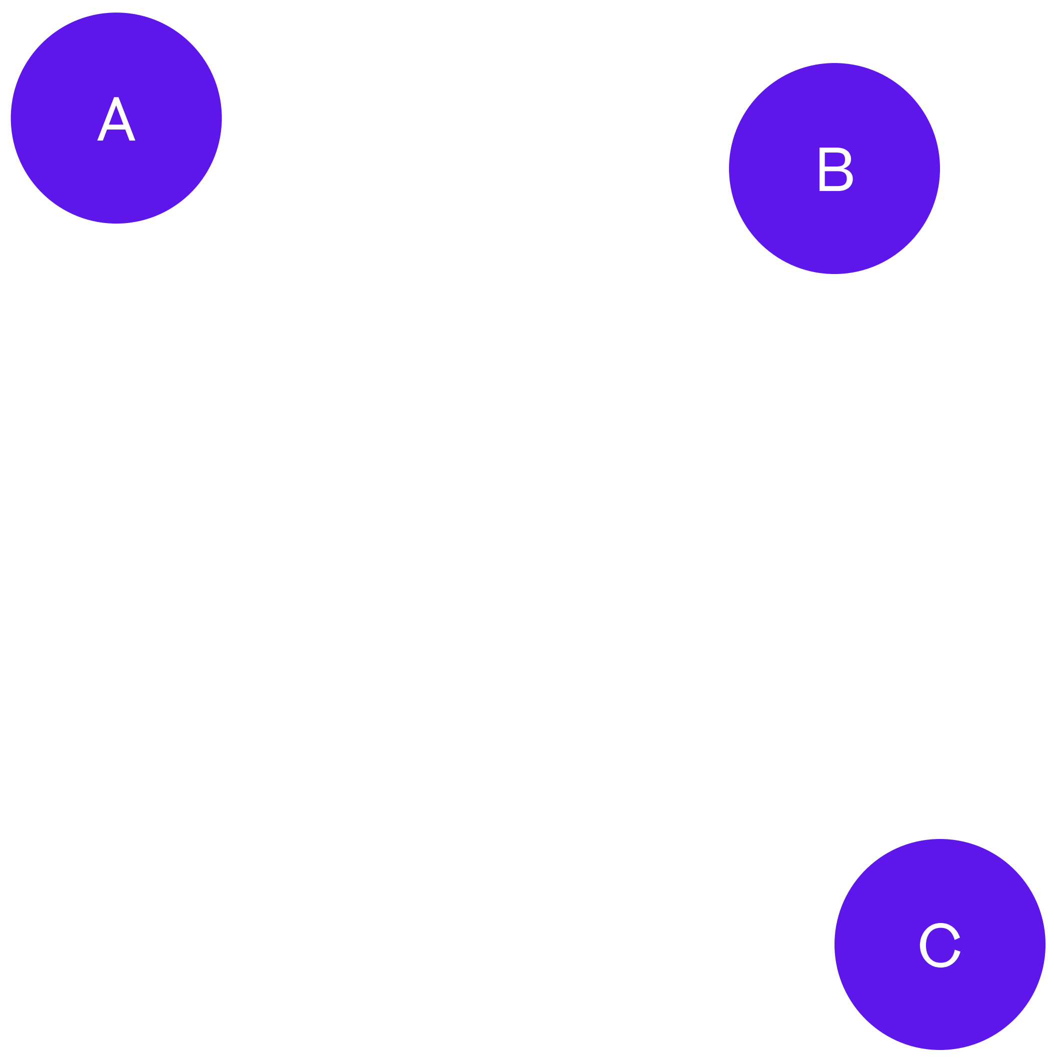
Application: Suivi épidémiologique

Scénario: Propagation du virus sur une vingtaine d’individus

Application: Suivi épidémiologique

Scénario: Propagation du virus sur une vingtaine d’individus

Sommets: individus



Application: Suivi épidémiologique

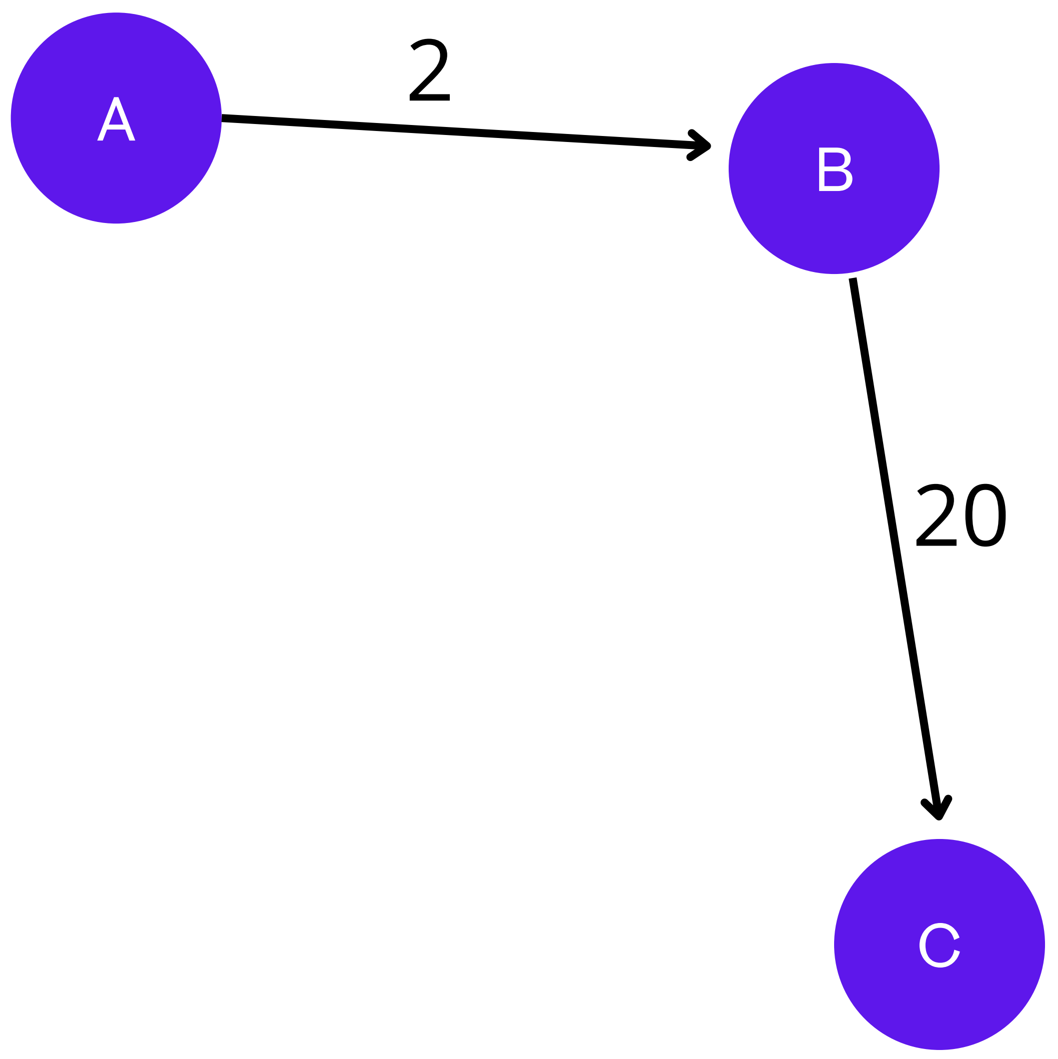
Scénario: Propagation du virus sur une vingtaine d’individus

Sommets: individus

Poids: “Résistance à la contamination”

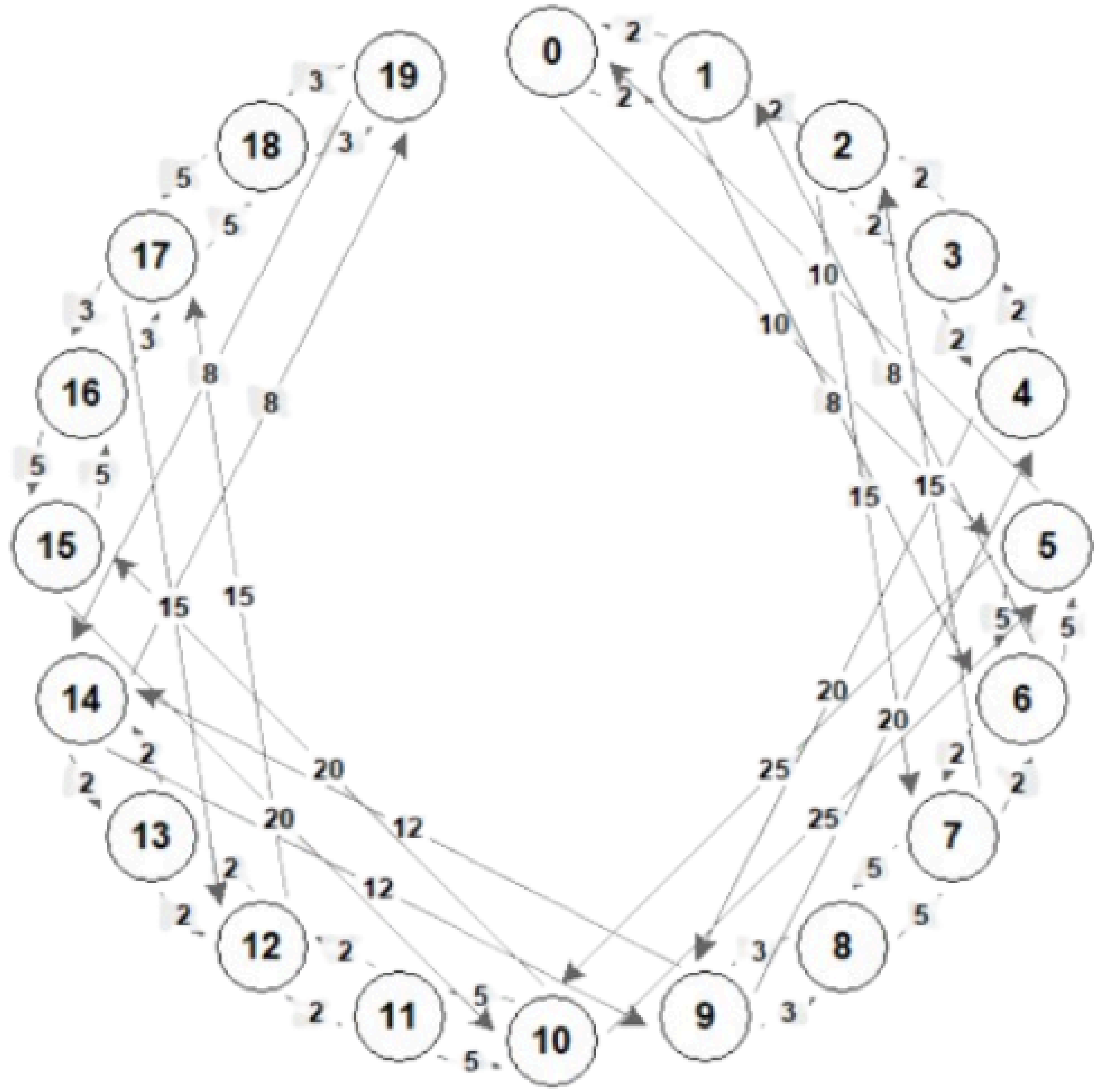
Poids 2 = Famille (Risque fort)

Poids 20 = Travail (Risque faible.



Resultat

Scénario : Transmission du Patient 0 au Patient 19.

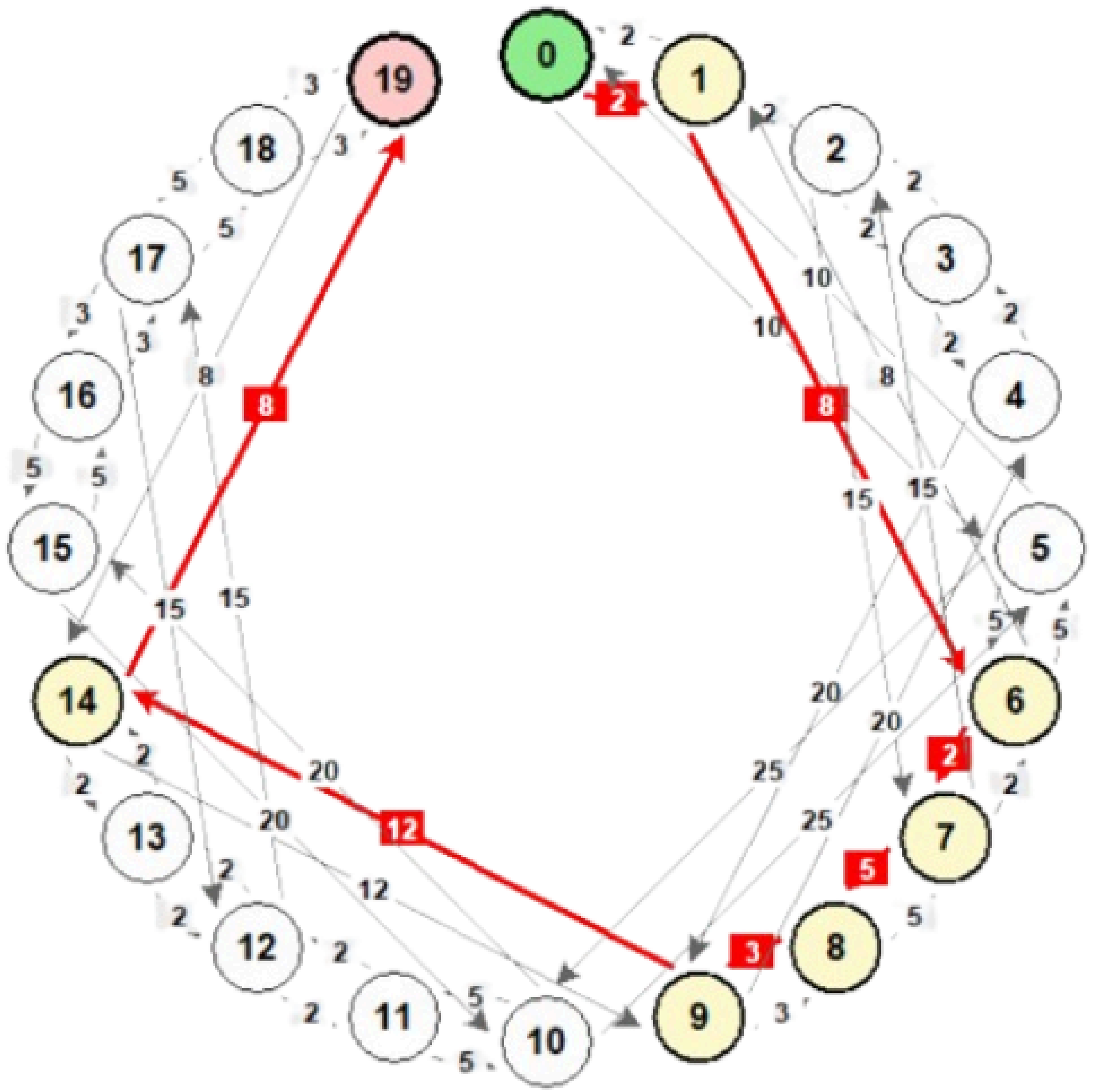


Resultat

Scénario : Transmission du Patient 0 au Patient 19.

Le virus contourne le travail pour passer par des "Super-Spreaders" (ponts sociaux).

Chemin : 0 → 1 → 6 → 7 → 8 → 9 → 14 → 19



Conclusion