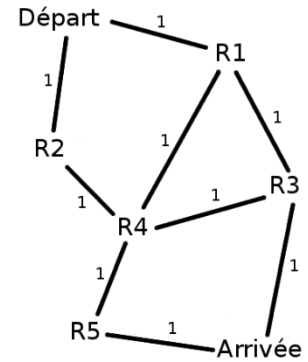


Le principe reste le même que l'algorithme de Bellman-Ford (logique puisque le protocole utilise cet algo !).

Contrairement au graphe précédent qui avait des distances différentes entre deux nœuds, ici la distance est de 1 (un saut).



Prenons en exemple le réseau ci-contre.

	Dep.	R1	R2	R3	R4	R5	Arr.
Distance avec Dep.	0	∞	∞	∞	∞	∞	∞
Routeur précédent	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

Au départ, la distance entre le routeur de départ et tous les autres routeurs est noté ∞ .

A chaque "tour", pour chaque routeur, on cherchera un plus court chemin parmi les routeurs adjacents.

Pour cet exemple, nous décrirons moins les étapes. Pour plus de détails, revoir la partie précédente sur l'algorithme de Bellman-Ford.

Tour n°1

Tous les routeurs sont mis à jour :

- Les routeurs **R1** et **R2** ont un plus court chemin.
- Les routeurs **R3**, **R4**, **R5** et **Arrivée** n'ont pas de chemin plus court.

	Dep.	R1	R2	R3	R4	R5	Arr.
Distance avec Dep.	0	1	1	∞	∞	∞	∞
Routeur précédent	\emptyset	Dep.	Dep.	\emptyset	\emptyset	\emptyset	\emptyset

Tour n°2

Tous les routeurs sont mis à jour :

- Les routeurs **R3** et **R4** ont un plus court chemin.
- Les routeurs **R1**, **R2**, **R5** et **Arrivée** n'ont pas de chemin plus court.

	Dep.	R1	R2	R3	R4	R5	Arr.
Distance avec Dep.	0	1	1	2	2	∞	∞
Routeur précédent	∅	Dep.	Dep.	R1	R1	∅	∅

Note : Le routeur R4 peut avoir R1 ou R2 comme routeurs précédents.

Tour n°3

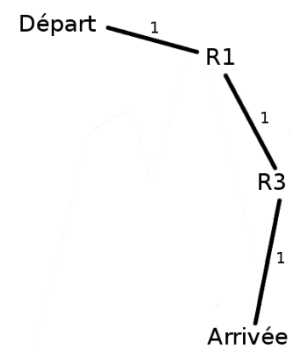
Tous les routeurs sont mis à jour :

- Les routeurs **R5** et **Arrivée** ont un plus court chemin.
- Les routeurs **R1**, **R2**, **R3** et **R4** n'ont pas de chemin plus court.

	Dep.	R1	R2	R3	R4	R5	Arr.
Distance avec Dep.	0	1	1	2	2	3	3
Routeur précédent	∅	Dep.	Dep.	R1	R1	R4	R3

Fin

Le chemin le plus court entre les nœuds Départ et Arrivée est **Départ-R1-R3-Arrivée** avec une distance de 3.



A noter que ce réseau possède plusieurs chemins plus courts. Les routeurs prennent le premier de ces chemins.