A MOULARD Valentin

7 14

9

B 2 F

C

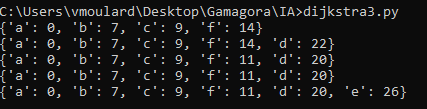
15 11 9

D 6 E

L’implémentation de l’algorithme de Dijkstra a été faite en Python.

C’est un algorithme récursif, qui a chaque nœud parcouru calcul la distance totale du nœud courant au nœud de départ. Le graphe fourni en paramètre de l’algorithme est un dictionnaire des pères de chaque nœud. L’algorithme prend en paramètre ce dictionnaire, le nœud de départ et d’arrivée. L’algorithme maintient une liste à jour des meilleures distances actuelles entre le nœud de départ et les sommets du graphe. Voici un tableau de Dijkstra montrant les différents états de cette liste des distances au fil de son exécution.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Etat | A | B | C | D | E | F | Next step |
| Initialisation | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | A |
| A | 0 | 7 | 9 | ∞ | ∞ | 14 | B |
| B | 0 | 7 | 9 | 22 | ∞ | 14 | C |
| C | 0 | 7 | 9 | 20 | ∞ | 11 | F |
| F | 0 | 7 | 9 | 20 | ∞ | 11 | D |
| D | 0 | 7 | 9 | 20 | 26 | 11 | E |
| E | 0 | 7 | 9 | 20 | 26 | 11 | / |



L’algorithme renvoie ensuite une liste contenant le chemin le plus court entre le nœud de départ et le nœud d’arrivée voulu. Ce processus est aussi récursif. Il part du nœud d’arrivée et remonte les nœuds père qui constitue le plus court chemin.

Dans la capture suivante, nous avons les plus courts chemins de l’exemple sur la page Wikipédia, pour le graphe présenté plus haut et aussi le chemin le plus court pour le même graphe mais cette fois ci non orienté. L’algorithme de Dijkstra marche pour les graphe non orienté et orienté.

