

Algorithme de Dijkstra

I. Rougier

Algorithme de Dijkstra

Algorithme

- **Etape 1** (Initialisation) :
 - Placer tous les sommets du graphe dans la 1^{ère} ligne d'un tableau.
 - Sur La 2^{ème} ligne, écrire le coefficient 0 pour le sommet de départ et les coefficient ∞ pour tous les autres sommets

Algorithme de Dijkstra

Algorithme

- **Etape 1** (Initialisation) :

- Placer tous les sommets du graphe dans la 1^{ère} ligne d'un tableau.
- Sur La 2^{ème} ligne, écrire le coefficient 0 pour le sommet de départ et les coefficient ∞ pour tous les autres sommets

Algorithme de Dijkstra

Algorithme

- **Etape 1** (Initialisation) :
 - Placer tous les sommets du graphe dans la 1^{ère} ligne d'un tableau.
 - Sur La 2^{ème} ligne, écrire le coefficient 0 pour le sommet de départ et les coefficient ∞ pour tous les autres sommets

Algorithme de Dijkstra

Algorithme

- **Etape 1** (Initialisation) :
 - Placer tous les sommets du graphe dans la 1^{ère} ligne d'un tableau.
 - Sur La 2^{ème} ligne, écrire le coefficient 0 pour le sommet de départ et les coefficient ∞ pour tous les autres sommets

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2 (Traitement) :**

- Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
- Pour tous les sommets Y non rayés :
 - Si Y est adjacent à X alors :
 - $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
 - Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
 - Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
- Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2 (Traitement) :**
 - Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
 - Pour tous les sommets Y non rayés :
 - Si Y est adjacent à X alors :
 - $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
 - Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
 - Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
 - Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2** (Traitement) :
- Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
- Pour tous les sommets Y non rayés :
 - Si Y est adjacent à X alors :
 - $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
 - Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
 - Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
- Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2** (Traitement) :
- Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
- Pour tous les sommets Y non rayés :
- Si Y est adjacent à X alors :
 - $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
 - Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
- Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
- Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2** (Traitement) :
- Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
- Pour tous les sommets Y non rayés :
- Si Y est adjacent à X alors :
- $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
- Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
- Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
- Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2** (Traitement) :
- Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
- Pour tous les sommets Y non rayés :
- Si Y est adjacent à X alors :
- $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
- Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
- Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
- Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2** (Traitement) :
- Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
- Pour tous les sommets Y non rayés :
- Si Y est adjacent à X alors :
- $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
- Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
- Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
- Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 2** (Traitement) :
- Repérer le sommet X de coefficient minimal : rayer la colonne sous ce sommet
- Pour tous les sommets Y non rayés :
- Si Y est adjacent à X alors :
- $p = \text{coefficient}(X) + \text{poids}(\text{arête } X-Y)$
- Si $p < \text{coefficient}(Y)$ alors remplacer le coefficient de Y par p ; sinon recopier le coefficient(Y).
- Si Y n'est pas adjacent à X alors recopier le coefficient(Y)
- Tant qu'il reste des colonnes non rayés, reprendre l'étape 2.

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 3 (Sortie) :**
 - Le poids de la chaîne la plus courte est le dernier nombre dans la colonne du sommet d'arrivée.
 - Pour connaître cette chaîne, on remonte à l'envers : partir du dernier nombre de la colonne du sommet. Rechercher l'étape de sa modification : prendre le sommet qui a été barré à cette étape. Procéder de même jusqu'à remonter au coefficient "0".

Algorithme de Dijkstra (suite)

Algorithme

- **Etape 3 (Sortie) :**
- Le poids de la chaîne la plus courte est le dernier nombre dans la colonne du sommet d'arrivée.
- Pour connaître cette chaîne, on remonte à l'envers : partir du dernier nombre de la colonne du sommet. Rechercher l'étape de sa modification : prendre le sommet qui a été barré à cette étape. Procéder de même jusqu'à remonter au coefficient "0".

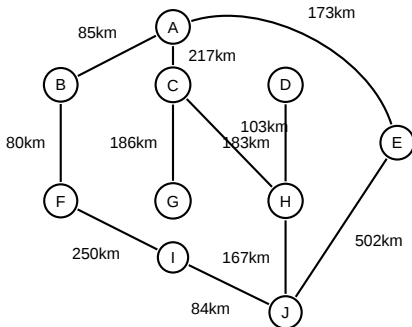
Algorithme de Dijkstra (suite)

Algorithme

- **Etape 3 (Sortie) :**
- Le poids de la chaîne la plus courte est le dernier nombre dans la colonne du sommet d'arrivée.
- Pour connaître cette chaîne, on remonte à l'envers : partir du dernier nombre de la colonne du sommet. Rechercher l'étape de sa modification : prendre le sommet qui a été barré à cette étape. Procéder de même jusqu'à remonter au coefficient "0".

Exemple

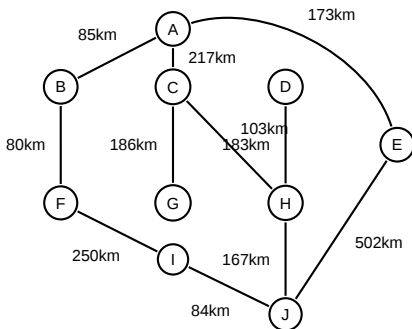
On considère le graphe suivant, qui représente les distances entre plusieurs villes.



On souhaite aller de la ville A à la ville J par le plus court chemin.

Exemple

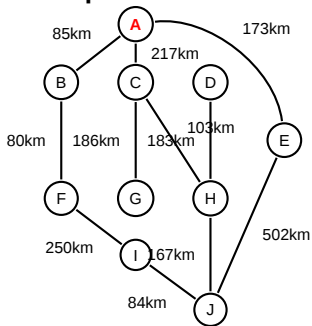
On considère le graphe suivant, qui représente les distances entre plusieurs villes.



On souhaite aller de la ville A à la ville J par le plus court chemin.

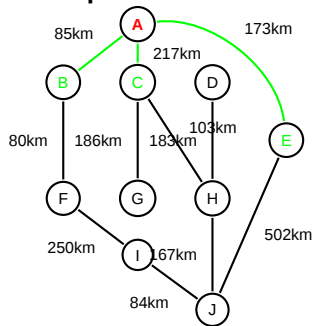
Exemple

Etape 1

[illegible]

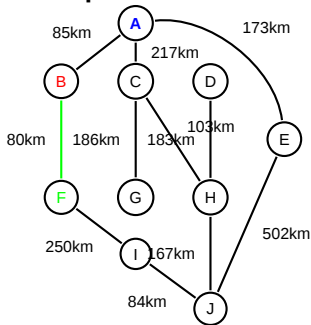
Example

Etape 2

[illegible]

Exemple

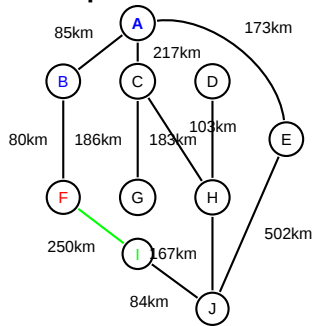
Etape 2



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞

Example

Etape 2



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞

Exemple

Etape 2

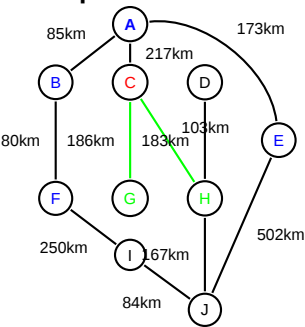
Diagram illustrating a network structure with 10 nodes (A, B, C, D, E, F, G, H, I, J) and weighted edges. The edges and their weights are:

- A-B: 85km
- A-C: 217km
- A-E: 173km
- B-F: 80km
- C-G: 186km
- C-H: 183km
- D-H: 103km
- F-I: 250km
- G-I: 167km
- H-J: 84km
- I-J: 84km
- E-J: 502km (highlighted in green)

A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675

Exemple

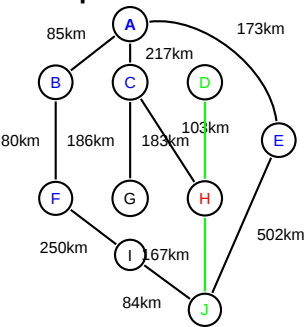
Etape 2



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675

Exemple

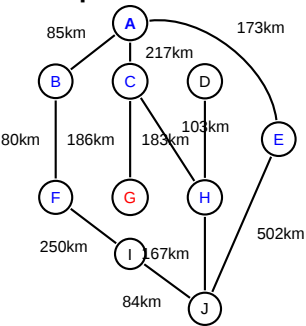
Etape 2



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487

Exemple

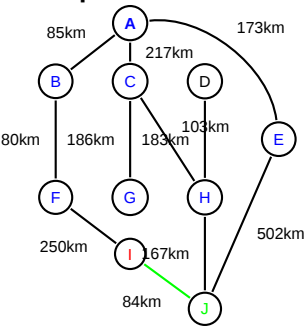
Etape 2



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487
			503					415	487

Exemple

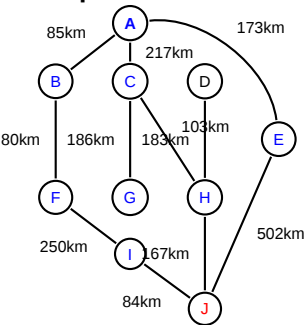
Etape 2



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487
			503					415	487
			503					499	487

Exemple

Etape 2



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487
			503					415	487
			503						487
			503						487

Exemple

Etape 2

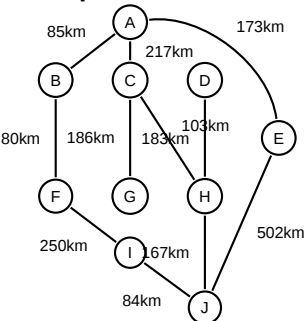
Diagram illustrating a weighted undirected graph with 10 nodes (A, B, C, D, E, F, G, H, I, J) and 12 edges. The edges and their weights are:

- A-B: 85km
- A-C: 217km
- A-E: 173km
- B-F: 80km
- C-G: 186km
- C-H: 183km
- D-H: 103km
- E-H: 502km
- F-I: 250km
- G-I: 167km
- I-J: 84km
- H-J: 167km

[illegible]

Exemple

Etape 3

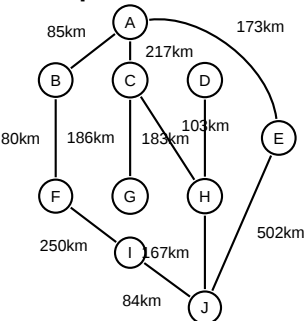


A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			∞	320	415	675
			503			403		415	487
			503					415	487
			503						499
									487

Le plus court chemin pour aller de A vers J est de 487 km.

Exemple

Etape 3

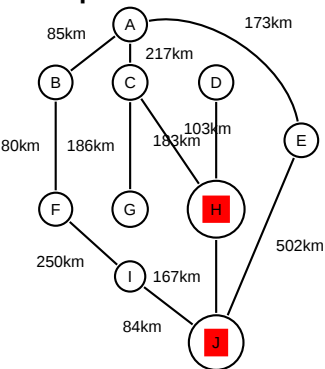


A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487
			503					415	487
			503						499
									487

Le plus court chemin pour aller de A vers J est de 487 km.

Exemple

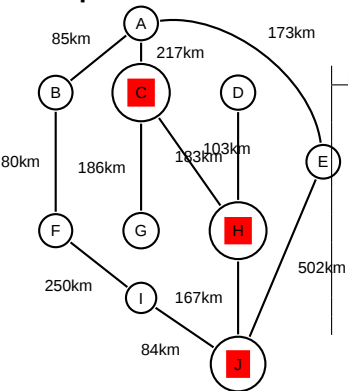
Etape 3



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487
			503					415	487
			503						499
			503						487

Exemple

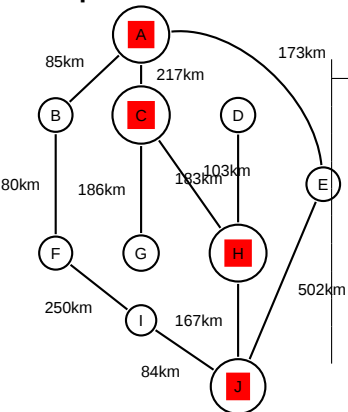
Etape 3



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
		217	∞			∞	∞	415	675
			∞			403	320	415	487
			∞			403		415	487
			503					415	487
			503						487
			503						487
			503						487

Exemple

Etape 3

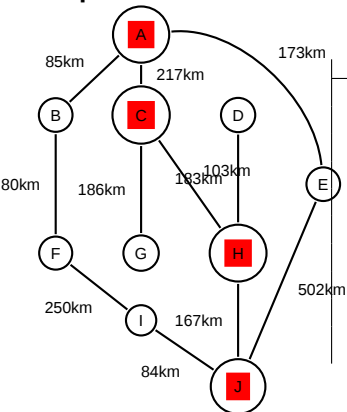


A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487
			503					415	487
			503						499
			503						48

La plus courte chaîne est alors A-C-H-J.

Exemple

Etape 3



A	B	C	D	E	F	G	H	I	J
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
	85	217	∞	173	∞	∞	∞	∞	∞
		217	∞	173	165	∞	∞	∞	∞
		217	∞	173		∞	∞	415	∞
		217	∞			∞	∞	415	675
			∞			403	320	415	675
			503			403		415	487
			503					415	487
			503						499
			503						48

La plus courte chaîne est alors A-C-H-J.



FIN