

C14 Plus court chemin dans un graphe

1. File de priorité

Définition

Une file de priorité est une structure de données, :

- qui permet d'*enfiler* (ajouter) des éléments et de *défiler* (retirer) des éléments (comme une file classique).

C14 Plus court chemin dans un graphe

1. File de priorité

Définition

Une file de priorité est une structure de données, :

- qui permet d'*enfiler* (ajouter) des éléments et de *défiler* (retirer) des éléments (comme une file classique).
- mais de plus, chaque élément enfilé possède une priorité.

C14 Plus court chemin dans un graphe

1. File de priorité

Définition

Une file de priorité est une structure de données, :

- qui permet d'*enfiler* (ajouter) des éléments et de *défiler* (retirer) des éléments (comme une file classique).
- mais de plus, chaque élément enfilé possède une priorité.
- lorsqu'on défile un élément, on enlève l'élément le plus prioritaire (et donc pas forcément le premier enfilé comme dans une file classique).

Exemple

On convient de noter les éléments par des lettres et leur priorité par un entier. Par exemple $(3, X)$ est l'élément X avec priorité 3 et on considère que la priorité maximale est *la plus petite priorité*. Représenter l'évolution de la file de priorité et donner les éléments extraits lors de l'exécution des opérations suivantes :

Enfiler $(5, A)$, Enfiler $(2, B)$, Enfiler $(4, C)$, Enfiler $(1, D)$, Enfiler $(3, E)$, Enfiler $(6, F)$, Défiler, Défiler, Enfiler $(2, G)$, Défiler.

C14 Plus court chemin dans un graphe

1. File de priorité

Implémentation avec une liste

Pour implémenter une file de priorité, on peut utiliser une liste de Python :

- l'opération enfiler correspond à l'ajout d'un élément à la fin de la liste (avec `append`).
- l'opération défiler correspond à la recherche de l'indice de l'élément de priorité maximale, puis à sa suppression de la liste (avec `pop`).

Exercice

- 1 Ecrire la fonction `enfile` qui prend en paramètre une file de priorité (une liste) et un élément (p, x) et qui ajoute l'élément à la file.
- 2 Ecrire la fonction `defile` qui prend en paramètre une file de priorité (une liste) et qui retire l'élément de priorité maximale de la file et le renvoie. On pourra écrire au préalable une fonction `indice_prio` qui renvoie l'indice de l'élément de plus petite priorité
- 3 Quelle est la complexité de ces deux fonctions ?

C14 Plus court chemin dans un graphe

1. File de priorité

Utilisation du module `heapq`

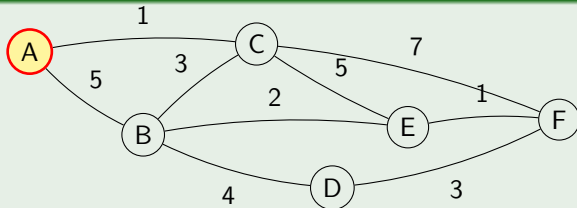
Le module `heapq` de Python permet de manipuler des files de priorité de manière efficace (les complexité des deux opérations sont logarithmiques)

- Si `f` est une liste vide, `heapify(f)` permet de transformer `f` en une file de priorité.
- La fonction `heappush` permet d'enfiler un élément dans la file.
- La fonction `heappop` permet de défiler l'élément de priorité maximale.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

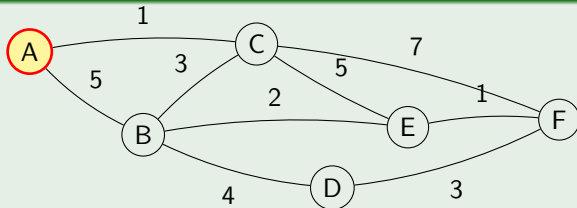


A	B	C	D	E	F	

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

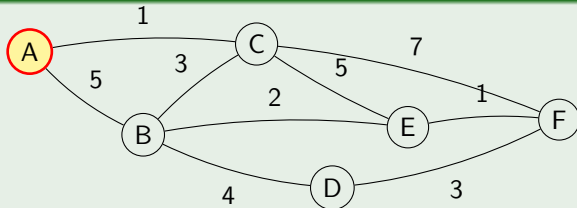


A	B	C	D	E	F	
(0,A)						

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

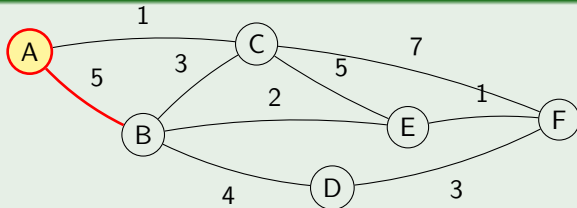


A	B	C	D	E	F	
(0,A)						A
✓						
✓						
✓						
✓						
✓						

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

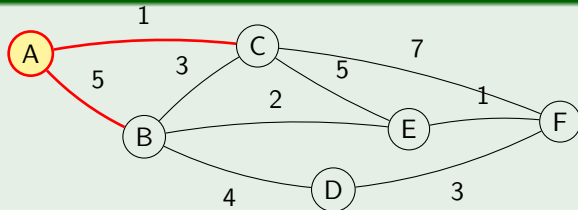


A	B	C	D	E	F	
(0,A)	(5,B)					A
✓						
✓						
✓						
✓						
✓						

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

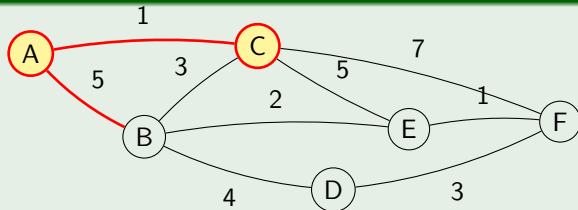


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓						
✓						
✓						
✓						
✓						

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

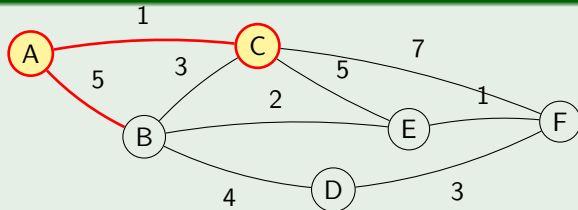


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓		(1,C)				
✓						
✓						
✓						
✓						

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

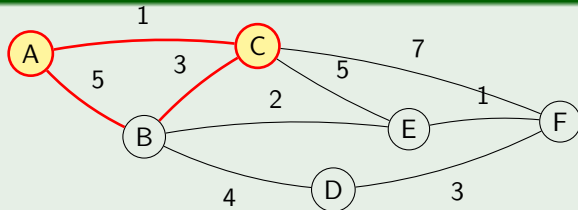


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓		(1,C)				C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

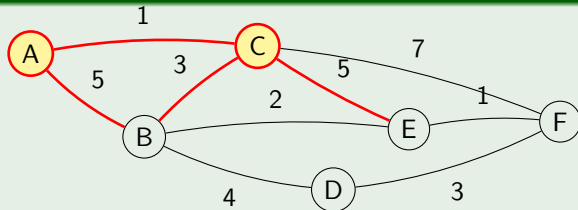


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)				C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

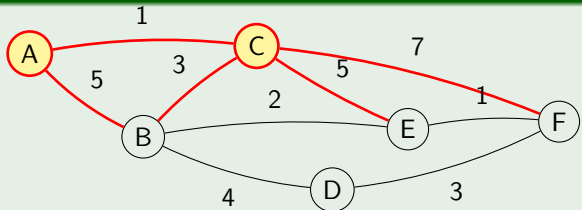


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)		C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

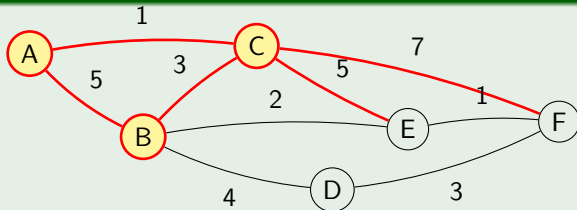


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓		✓				
✓		✓				
✓		✓				
✓		✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

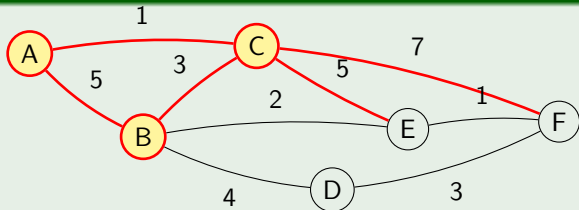


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓				
✓		✓				
✓		✓				
✓		✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

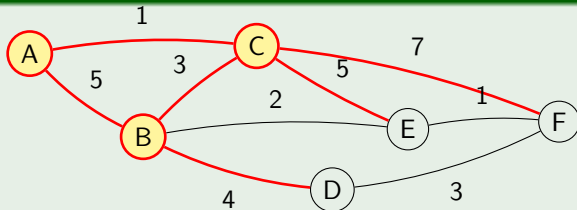


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓				B
✓	✓	✓				
✓	✓	✓				
✓	✓	✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

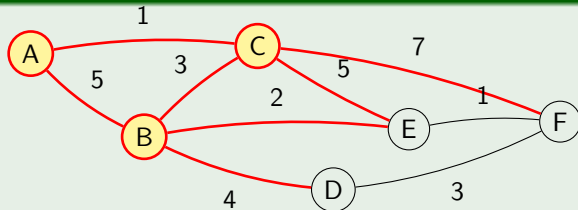


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)		(8,F)	B
✓	✓	✓				
✓	✓	✓				
✓	✓	✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

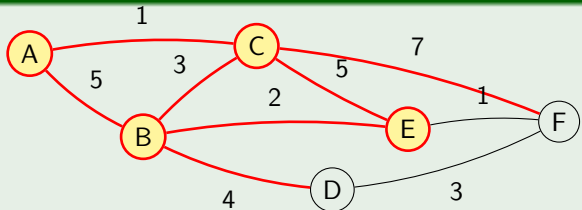


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓				
✓	✓	✓				
✓	✓	✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

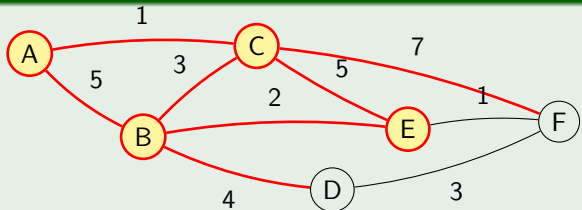


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	
✓	✓	✓				
✓	✓	✓				

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

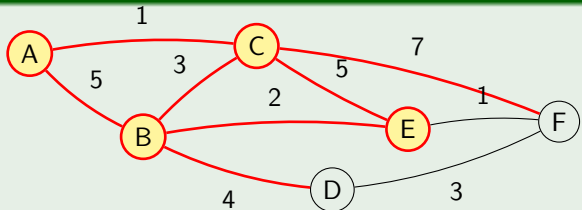


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓		✓		
✓	✓	✓		✓		

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

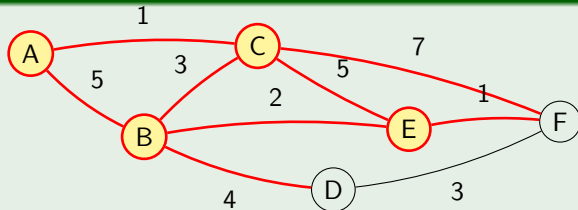


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓		✓		
✓	✓	✓		✓		

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

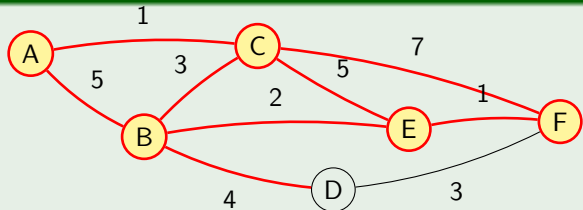


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓		✓		
✓	✓	✓		✓		

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

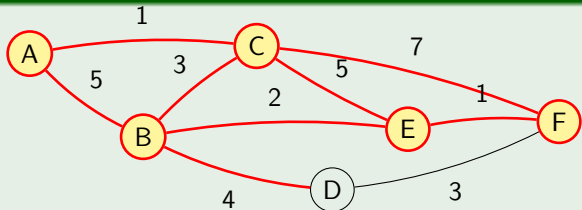


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓		✓	(7,F)	
✓	✓	✓		✓	✓	

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

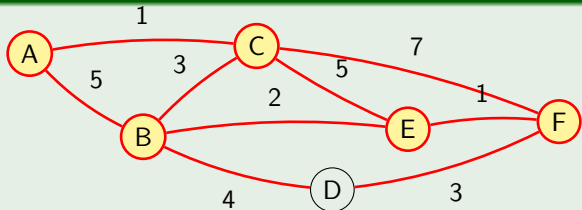


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓		✓	(7,F)	F
✓	✓	✓		✓	✓	

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

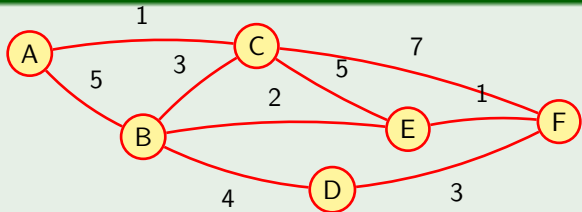


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓	(8,D)	✓	(7,F)	F
✓	✓	✓		✓	✓	

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple

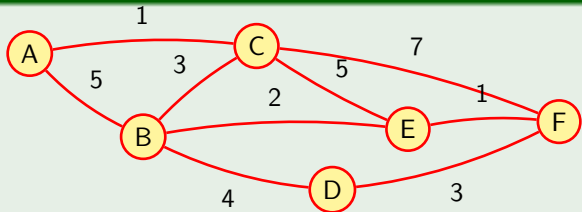


A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓	(8,D)	✓	(7,F)	F
✓	✓	✓	8 (B)	✓	✓	

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Algorithme de Dijkstra : exemple



A	B	C	D	E	F	
(0,A)	(5,B)	(1,C)				A
✓	(4,B)	(1,C)		(6,E)	(8,F)	C
✓	(4,B)	✓	(8,D)	(6,E)	(8,F)	B
✓	✓	✓	(8,D)	(6,E)	(7,F)	E
✓	✓	✓	(8,D)	✓	(7,F)	F
✓	✓	✓	8 (B)	✓	✓	D

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Principe

- L'algorithme de Dijkstra recherche le plus court chemin depuis un sommet de départ s vers *tous les autres* sommets du graphe.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Principe

- L'algorithme de Dijkstra recherche le plus court chemin depuis un sommet de départ s vers *tous les autres* sommets du graphe.
- Les poids sont supposés positifs.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Principe

- L'algorithme de Dijkstra recherche le plus court chemin depuis un sommet de départ s vers *tous les autres* sommets du graphe.
- Les poids sont supposés **positifs**.
- C'est une adaptation du parcours en largeur, dans laquelle la file des sommets en attente de traitement est une **file de priorité**.

Principe

- L'algorithme de Dijkstra recherche le plus court chemin depuis un sommet de départ s vers *tous les autres* sommets du graphe.
- Les poids sont supposés **positifs**.
- C'est une adaptation du parcours en largeur, dans laquelle la file des sommets en attente de traitement est une **file de priorité**.
- La priorité est la distance minimale trouvée pour le moment depuis le sommet de départ.

Idéalement, on doit donc disposer d'une file de priorité où la mise à jour de la priorité d'un élément est disponible. Pour notre implémentation, par souci de simplicité, on adoptera une file de priorité « traditionnelle » et on enfilera plusieurs fois le même sommet (avec des priorités différentes). Et on utilisera un tableau de booléens afin de ne pas traiter deux fois le même sommet. Cela a des conséquences en terme de complexité mais non significatives à notre niveau.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Description de l'algorithme

- 1 Initialiser la file de priorité avec le sommet de départ s et la distance 0.
- 2
- 3 La distance finale vers chaque sommet est donnée par le tableau des distances.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Description de l'algorithme

- 1 Initialiser la file de priorité avec le sommet de départ s et la distance 0.
- 2 Tant que la file de priorité n'est pas vide :
 - 3 La distance finale vers chaque sommet est donnée par le tableau des distances.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Description de l'algorithme

- ① Initialiser la file de priorité avec le sommet de départ s et la distance 0.
- ② Tant que la file de priorité n'est pas vide :
 - ① Défiler le sommet u de priorité minimale.
- ③ La distance finale vers chaque sommet est donnée par le tableau des distances.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

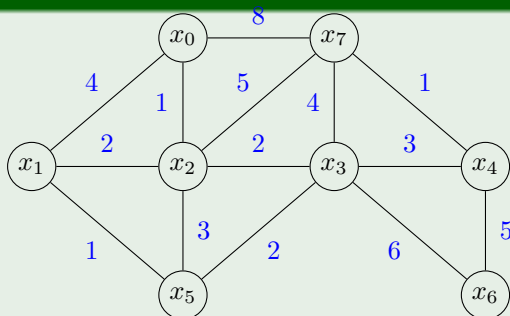
Description de l'algorithme

- ① Initialiser la file de priorité avec le sommet de départ s et la distance 0.
- ② Tant que la file de priorité n'est pas vide :
 - ① Défiler le sommet u de priorité minimale.
 - ② Si il n'a encore été traité le marquer comme traité et enfiler chacun de ses fils en leur donnant comme priorité celle de u ajouté à la pondération de l'arc les joignant.
- ③ La distance finale vers chaque sommet est donnée par le tableau des distances.

C14 Plus court chemin dans un graphe

2. Algorithme de Dijkstra

Exemple



Déterminer les plus courtes distances depuis le sommet x_0 vers les autres sommets du graphe en utilisant l'algorithme de Dijkstra.