ALGORITHME DE DIJKSTRA

TI-83 Premium CE

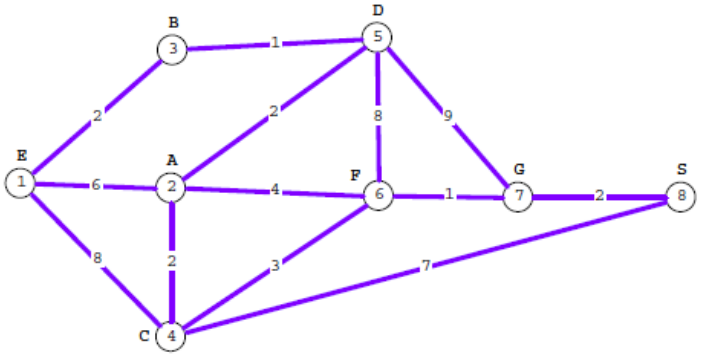
1. Objectifs

Certains problèmes consistent à chercher, entre deux points donnés d'un graphe, le parcours de poids minimal (durée, coût, distance).
E.W. Dijkstra (1930-2002) a proposé en 1959 un algorithme qui permet de déterminer le plus court chemin entre deux sommets d'un **graphe connexe pondéré (orienté ou non) dont le poids lié aux arêtes (ou arcs) est positif ou nul.**
Nous allons implémenter l'**algorithme de Dijkstra**, adapté à la recherche de ce parcours.

2. Énoncé

Un livreur prépare sa tournée. II doit visiter un certain nombre de ses clients nommés A, B, C, D, F et G en partant de E pour arriver en S.
Les liaisons possibles sont représentées sur le graphe G1 ci-contre pondéré par les durées, en minutes, des trajets.

Quel chemin doit-il emprunter pour minimiser la durée totale du trajet de E à S ?



3. Conduite de l'activité

Le graphe donné en exemple possède 8 sommets il est donc d'ordre 8 mais l'activité sera prévue pour s'adapter à des graphes d'ordre *n* où *n* est un entier supérieur ou égal à 3.

1) Saisie du graphe

Le graphe sera représenté par une matrice carré d'ordre *n* notée A. Les listes et matrices de la calculatrice ne permettent que de traiter des données numériques donc les sommets du graphe sont numérotés. Ils seront associés à un numéro de ligne (et de colonne) en plaçant le sommet de départ du trajet en 1 et le sommet d'arrivée en *n*.

Sommet	**E**	A	B	C	D	F	G	**S**
Ligne ou colonne	1	2	3	4	5	6	7	8

L'élément a_{ij} de la ligne *i* et de la colonne *j* sera le poids de l'arête reliant le sommet *i* au sommet *j* si elle existe sinon 0.
Par exemple $a_{12} = a_{21} = 6$ c'est l'arête reliant A et E.
La calculatrice note une variable matrice entre crochets avec une lettre majuscule de A à J donc [A] ici. L'élément a_{ij} se note **[A] (I,J).**
L'éditeur de matrice est la solution simple pour saisir les données du graphe.

i. Accès à l'éditeur avec la touche [matrice] en mode édition (**ÉDIT**).
ii. Définir la dimension de la matrice A.
iii. Saisir ses coefficients.
iv. Sortir de l'éditeur avec la touche « quitter » classique.

$$A = \begin{pmatrix} 0 & 6 & 2 & 8 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 2 & 2 & 4 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 8 & 2 & 0 & 0 & 0 & 3 & 0 & 7 \\ 0 & 2 & 1 & 0 & 0 & 8 & 9 & 0 \\ 0 & 4 & 0 & 3 & 8 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 9 & 1 & 0 & 2 \\ 0 & 0 & 0 & 7 & 0 & 0 & 2 & 0 \end{pmatrix}$$

2) Programmation de l'algorithme de Dijkstra

Il faudra commencer par relire et bien comprendre l'algorithme vu en cours avant de la programmer. Des choix s'imposent :

- Le tableau de sortie de l'algorithme sera sous forme d'une matrice carrée B d'ordre 8.
- Choisir un représentant numérique pour le **coefficient ∞,** (le choix de « -1 » est retenu ici)
- Choisir un représentant numérique pour les **sommets sélectionnés,** (le choix de « -2 » est retenu ici)
- La recherche de **coefficient minimal** sera faite par un sous-programme **MINL**
- La **chaîne de poids minimal** sera gérée dans la liste $\mathbf{L_1}$ de la calculatrice.

a) Programmation de MINL

Prérequis : La matrice **B** carrée d'ordre **N** est supposée contenir sur sa ligne S des valeurs dont au moins une est strictement positive. L'algorithme cherche, sur la ligne S, la plus petite valeur et sa colonne qui seront stockées respectivement dans les variables globales **T** et **L**.

Algorithme MINL		Sous-programme MINL
Variables locales: **I indice de colonne.** **I prend la valeur 2** **Tant que B(S,I) < 0** **I est augmenté de 1** **Fin Tant Que** **T prend la valeur B(S,I)** **L prend la valeur I**	**Pour I allant de L à N** **Si 0 < B(S,I) < T alors** **T prend la valeur B(S,I)** **L prend la valeur I** **Fin si** **Fin pour** **(afficher L et T pour la mise au point)**	**Créer** un nouveau programme nommé **MINL** avec la touche [prgm] en mode **NOUVEAU**. **Programmer** l'algorithme ci-contre.

b) Test de MINL

i. Éditer la matrice B suivante :

$$B = \begin{pmatrix} 0 & -1 & -1 & -1 & -1 & -1 & -1 \\ -2 & 4 & 2 & -1 & -1 & -1 & -1 \\ -2 & 4 & -2 & 7 & -1 & 5 & -1 \\ -2 & -2 & -2 & 7 & -1 & 5 & -1 \\ -2 & -2 & -2 & 7 & -1 & -2 & 13 \\ -2 & -2 & -2 & -2 & 8 & -2 & 12 \\ -2 & -2 & -2 & -2 & -2 & -2 & 12 \end{pmatrix}$$

ii. Initialiser les variables S à 2 et T à 0.
iii. Ajouter à la fin de MINL, l'affichage de L et T.
iv. Exécuter MINL
v. Recommencer avec S = 3 …etc.

Analyser le résultat

Pour S=2 :
La ligne 2 de B contient les valeurs ……
La plus petite valeur positive est ……..
Le programme donne : L = …… et T = …….
Est-ce correct ?.....

Recommencer pour S = 3 …etc.

NB : penser à retirer l'affichage de L et T.

c) Programmation de DIJKSTRA

Prérequis : Le graphe est stocké dans la matrice **A,** de ligne courante analysée **L**.
En sortie, la matrice **B,** de ligne courante **S**, contient le tableau de progression de l'algorithme.
La longueur minimale est dans la variable **T** et la chaine correspondante dans la liste $\mathbf{L_1}$.

Algorithme de DIJKSTRA	Programmation

Demander l'ordre **N** du graphe
Initialiser la matrice de sortie **B** à 0
Initialiser $\mathbf{L_1}$ à 0.
L prend la valeur 1, **T** la valeur 0 et **S** la valeur 2
Pour J allant de 2 à N ----------
B(1,J) prend la valeur -1
Fin pour

Tant Que S ≤ N ----------
B(S,1) et B(S,L) prennent la valeur -2 ----------
Pour J allant de 2 à N ----------
Si J≠L **alors** ----------
Si A(L,J)≠0 **alors** ----------
P prend la valeur A(L,J)+T ----------
Si B(S–1,J)=-2 **alors** ----------
[B](S,J) prend la valeur -2
Sinon
Si B(S–1,J) = -1 **ou** P < B(S–1,J) **alors** ----------
B(S,J) prend la valeur P et L_1(J) la valeur L
Sinon
B(S,J) prend la valeur B(S–1,J) ----------
Fin Si
Fin Si
Sinon

B(S,J) prend la valeur B(S–1,J) ----------
Fin Si
Fin Si
Fin Pour
Appeler MINL ----------
S est augmenté de 1 ----------
Fin Tant Que

Afficher le poids minimal T obtenu
Tant que L_1(N)≠0 ----------
Afficher L_1(N)
N prend la valeur L_1(N) ----------
Fin Tant Que

Liste $\mathbf{L_1}$

0	0	0	0	0	0	0	0

Matrice **B** (premières lignes)

0	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Ajouter les commentaires sur les pointillés pour justifier la traduction de l'algorithme vu en cours.

Programmer cet algorithme qui portera le nom DIJKSTRA

Compléter le tableau d'état pour
S = 2 et la matrice A du graphe (N = 8)
On a L = 1, T = 0

$$A = \begin{pmatrix} 0 & 6 & 2 & 8 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 2 & 2 & 4 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 8 & 2 & 0 & 0 & 0 & 3 & 0 & 7 \\ 0 & 2 & 1 & 0 & 0 & 8 & 9 & 0 \\ 0 & 4 & 0 & 3 & 8 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 9 & 1 & 0 & 2 \\ 0 & 0 & 0 & 7 & 0 & 0 & 2 & 0 \end{pmatrix}$$

J	2						
J ≠ L	V						
A(L,J) ≠ 0	F						
P	6						

On a maintenant (à compléter):

Liste $\mathbf{L_1}$

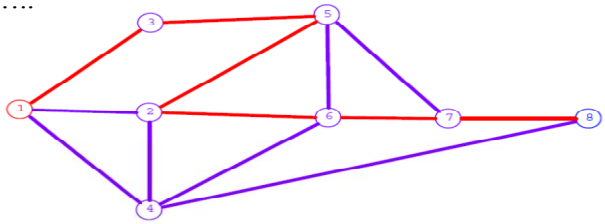
--	1	--	--	--	--	--	--

Matrice **B** (premières lignes)

0	-1	-1	-1	-1	-1	-1	-1
-2	6	--	--	--	--	--	--
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

d) Tests et utilisation

La matrice A est initialisée.
Exécuter le programme depuis la même touche « prgm ».
D'après l'affichage, on peut affirmer que le chemin le plus court de E à S est donc de …… **min**.
Il reste à interpréter la chaine affichée …… avec la correspondance des sommets ce qui donne le trajet : ……..



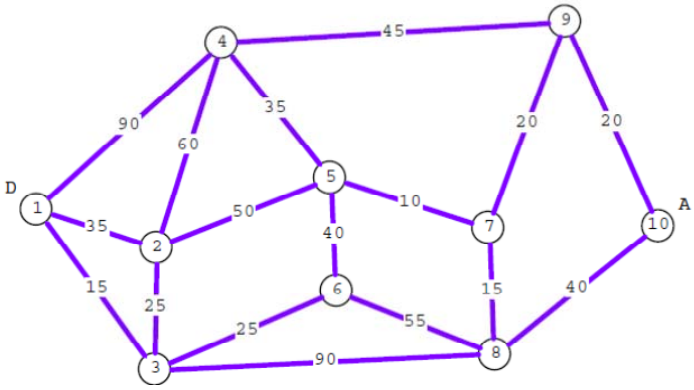
Le détail de traitement peut être observé en affichant la **matrice B** et la liste de chainage $\mathbf{L_1}$ toujours disponibles après exécution.

- Afficher la matrice B
- Compléter le tableau présentant l'exécution de l'algorithme comme demandé par le programme :

E	A	B	C	D	F	G	**S**
0	∞	∞	∞	∞	∞	∞	∞

3) Un autre graphe

D'après bac ES Antilles–Guyane-juin 2013.



Un guide de randonnée en montagne décrit les itinéraires possibles autour d'un pic rocheux. Les temps de parcours pour chacun des sentiers sont en minutes.

Déterminer l'itinéraire allant de D à A, le plus court en temps.

D	2	3	4	5	6	7	8	9	**A**

i. Sauvegarder la matrice A précédente dans la matrice C.
ii. Initialiser la nouvelle matrice A de dimension : .....
iii. Exécuter l'algorithme
iv. Compléter le tableau d'après la matrice B.

Le chemin le plus court de D à A est donc de : …… minutes.
La chaine affichée est : ……
Avec la correspondance des sommets, on obtient le trajet : ……