

ALGORITHME DE DIJKSTRA

Auteur : Marie-Laurence Brivezac

TI-83 Premium CE

Mots-clés : graphes, matrices, algorithme, programmation.

Fichiers associés : dijkstra_eleve.pdf, DIJKSTRA.8xp, MINL.8xp, [C].8xm, [D].8xm.

1. Objectifs

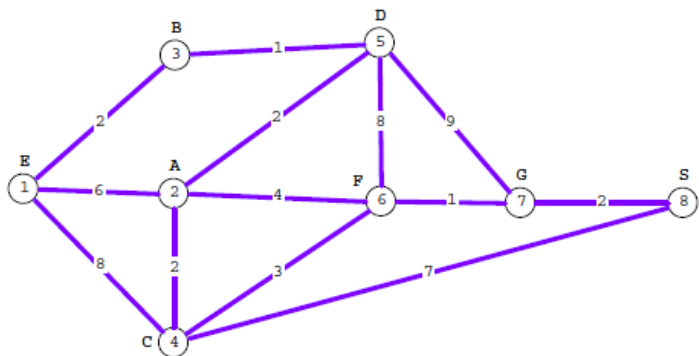
Certains problèmes consistent à chercher, entre deux points donnés d'un graphe, le parcours de poids minimal (durée, coût, distance). Nous allons implémenter l'**algorithme de Dijkstra**, adapté à la recherche de ce parcours, dans le cadre d'une classe de terminale ES spécialité mathématiques.

2. Énoncé

Un livreur prépare sa tournée. Il 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 ci-contre pondéré par les durées, en minutes, des trajets¹.

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



3. Commentaires

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 est positif ou nul** :

Début Placer tous les sommets du graphe dans la ligne 1 d'un tableau ;

Sur la ligne 2 : le coefficient 0 sous le point de départ et le **coefficient ∞** sous les autres sommets.

Sélectionner sur la ligne courante le sommet X de **coefficient minimal**;

Tant Qu'il reste des sommets non sélectionnés

Commencer une nouvelle ligne et rayer toutes les cases vides sous X en tirant un trait vertical.

Pour chaque sommet Y du graphe, adjacent à X

Calculer la somme p du poids courant et du poids de l'arête reliant X à Y

Si p est strictement inférieur au coefficient précédent de Y **alors**

Inscrire p dans la colonne « Y » **et noter** le sommet de **provenance**

Sinon

Recopier le coefficient précédent de Y

Fin Si

Fin Pour

Compléter la ligne par les coefficients de la ligne précédente.

Sélectionner le sommet X de **coefficient minimal** et commencer une nouvelle ligne.

Fin Tant Que

Fin **Afficher** le poids minimal obtenu et la chaîne de poids minimal.

¹ Le départ, l'arrivée et chacun des clients sont numérotés, pour permettre une exploitation à l'aide de la calculatrice.

Remarque : Cet algorithme s'applique dans le cas des graphes orientés, on remplace arêtes par arcs et on tient compte de l'orientation dans la matrice A qui ne sera plus symétrique.

4. 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 non nul supérieur ou égal à 3. Pour rester sur un temps de calcul raisonnable et un affichage convenable sur l'écran de la calculatrice, il ne faudra pas dépasser beaucoup plus de $n = 10$.

1) Saisie du graphe

Le graphe sera représenté par une matrice carré d'ordre n notée A. Les sommets du graphe 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

Les listes et matrices de la calculatrice ne permettent que de traiter des données numériques.

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.

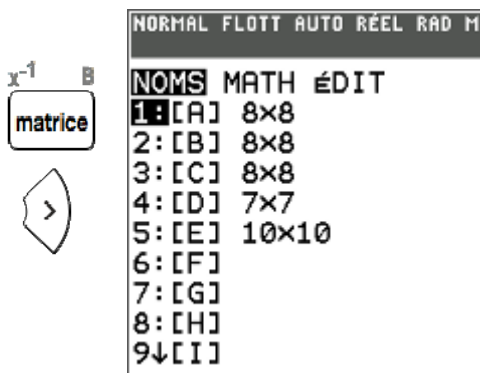
Par exemple $a_{12} = a_{21} = 6$ c'est l'arête reliant A et E.

La calculatrice note une variable matrice entre crochets avec une lettre majuscule de A à J donc [A] ici.

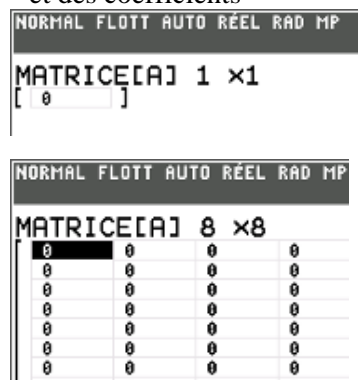
L'élément a_{ij} se note [A](I,J).

L'éditeur de matrice est la solution simple pour saisir les données du graphe.

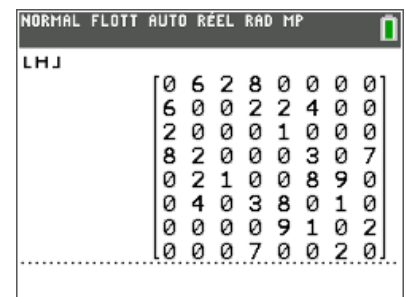
1. Accès à l'éditeur



2. Saisie de la dimension et des coefficients



3. Affichage de vérification



Il est à noter que la matrice est initialisée à 0 lors de sa création. Seuls les coefficients différents de zéro seront saisis. La sortie de l'éditeur se fait avec la touche « quitter » classique. Cette matrice pourra être modifiée de la même manière. On pourra stocker plusieurs graphes en utilisant les matrices de C à J. C'est la matrice [C].8xm donnée dans les fichiers de l'activité.

2) Programmation de l'algorithme

Quelques choix :

- Le tableau de sortie de l'algorithme sera sous forme d'une matrice carrée B d'ordre 8.
- le coefficient ∞ sera représenté par la valeur -1 et les sommets sélectionnés par la valeur -2.
- La recherche de coefficient minimal sera faite par un sous-programme MINL
- La chaîne de poids minimal sera gérée dans la liste L1 de la calculatrice.

La programmation peut être faite directement depuis la calculatrice à partir de l'éditeur de programmes accessible depuis la touche ou depuis le logiciel TI Connect CE.

Edition avec TI Connect CE

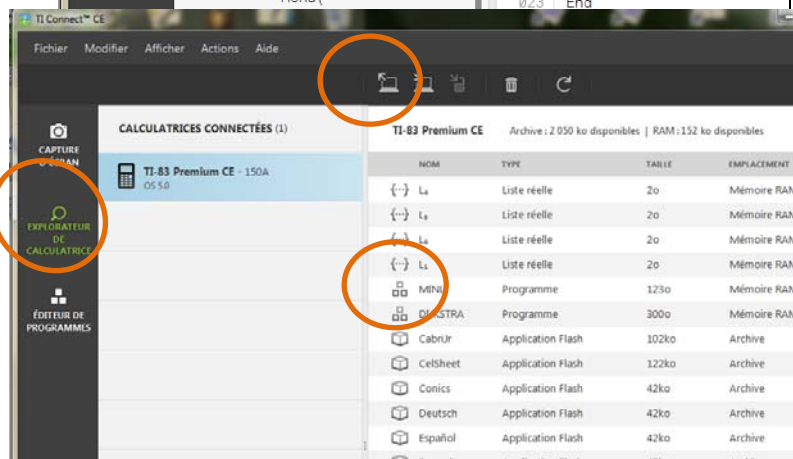
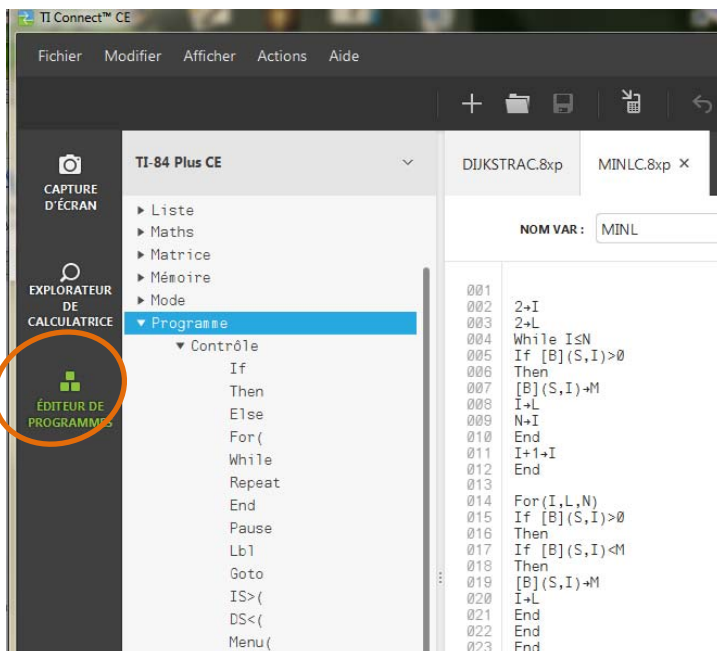
La calculatrice est connectée à l'ordinateur par le câble fourni avec la calculatrice (liaison par prise USB).

TI Connect CE possède un éditeur intégré plus confortable et la possibilité de gérer la communication avec la calculatrice et notamment sauvegarde et restauration des différentes données.

Les **fichiers** programme de la calculatrice sont stockés sur l'ordinateur avec l'extension **.8xp**.

Les **variables** comme les matrices **A** et **B** sont également accessibles, en fait l'état global de la calculatrice.

Des **captures d'écran** sont également possibles.



Edition sur calculatrice

1^{re} étape : Créer un nouveau programme, le nom est limité à 8 caractères majuscules.

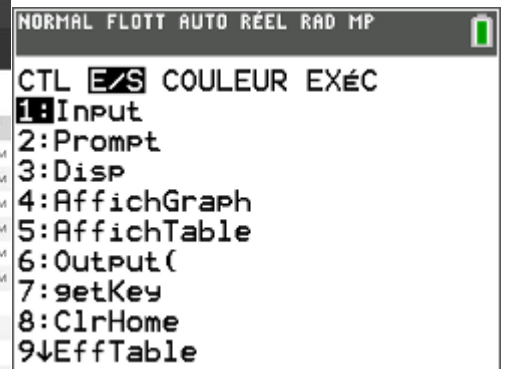
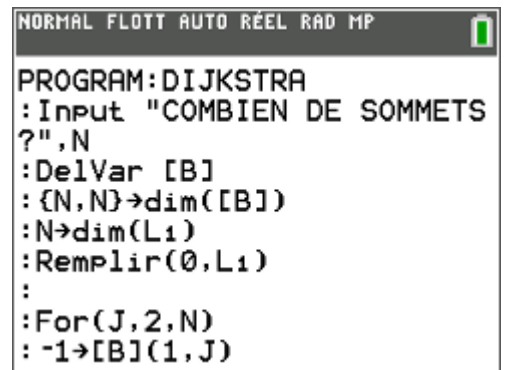
La suppression éventuelle de programmes se fait avec les touches :



(voir en fin de document)



2^e étape : l'édition, les instructions sont accessibles depuis la touche



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 chaîne correspondante dans la liste **L₁**.

Programme DIJKSTRA	Sous-programme MINL
<pre> Input "COMBIEN DE SOMMETS?",N DelVar [B] {N,N}→dim([B]) N→dim(L₁) Remplir(0,L₁) For(J,2,N) -1→[B](1,J) End 1→L:0→T:2→S While S≤N -2→[B](S,L) -2→[B](S,1) For(J,2,N) If J≠L Then If [A](L,J)≠0 Then [A](L,J)+T→P If [B](S-1,J)=-2 Then -2→[B](S,J) </pre>	<pre> Else If [B](S-1,J)=-1ou[B](S-1,J)>P Then P→[B](S,J) L→L₁(J) Else [B](S-1,J)→[B](S,J) End End Else [B](S-1,J)→[B](S,J) End End End prgmMINL S+1→S End ClrHome 22→X Disp "LONGUEUR:" Output(1,11,T) Disp "CHEMIN:" Output(2,X+1,N) While L₁(N)≠0 Output(2,X,"-") Output(2,X-1,L₁(N)) L₁(N)→N X-2→X End </pre>
	<p>Prérequis : La matrice B est supposée contenir sur sa ligne S des valeurs dont au moins une est strictement positive.</p> <p>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.</p> <pre> 2→I While [B](S,I)<0 I+1→I End [B](S,I)→T I→L For(I,L,N) If [B](S,I)>0et[B](S,I)<T Then [B](S,I)→T I→L End End DelVar I </pre>

3) Tests et utilisation pour le graphe de l'énoncé

la matrice **A** aura été saisie au préalable.

Le détail de traitement peut être observé en affichant la **matrice B** et la liste de chaînage **L₁** toujours disponibles après exécution.

Ces variables, n'étant pas indispensables en termes de résultat, ne sont pas affichées par le programme.

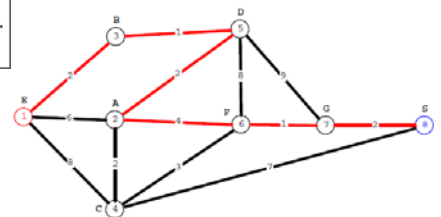


LB	J	1	2	3	4	5	6	7	8
0	-1	-1	-1	-1	-1	-1	-1	-1	-1
-2	6	2	8	-1	-1	-1	-1	-1	-1
-2	6	-2	8	3	-1	-1	-1	-1	-1
-2	5	-2	8	-2	11	12	-1	-1	-1
-2	-2	-2	7	-2	9	12	-1	-1	-1
-2	-2	-2	-2	-2	9	12	14	-1	-1
-2	-2	-2	-2	-2	-2	10	14	-1	-1
-2	-2	-2	-2	-2	-2	-2	12	-1	-1

Le chemin le plus court de E à S est donc de **12 min**.

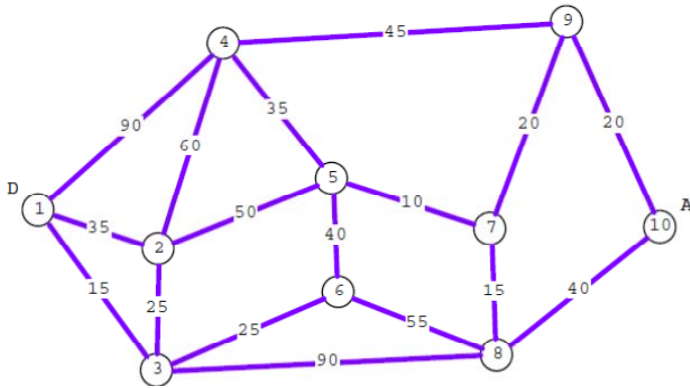
La chaîne affichée 1-3-5-2-6-7-8 avec la correspondance des sommets donne le trajet :

E-B-D-A-F-G-S.



4) Un autre graphe

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



On commencera par modifier la **matrice A** qui est d'ordre 10 maintenant. C'est la matrice **[D]**, $8 \times m$ donnée dans les fichiers de l'activité.

0	35	15	90	0	0	0	0	0	0
35	0	25	60	50	0	0	0	0	0
15	25	0	0	0	25	0	90	0	0
90	60	5	0	35	0	0	0	45	0
0	50	0	35	0	40	10	0	0	0
0	0	25	0	40	0	0	55	0	0
0	0	0	0	10	0	0	15	20	0
0	0	90	0	0	55	15	0	0	40
0	0	0	45	0	0	0	20	0	20
0	0	0	0	0	0	0	40	20	0

La **matrice B** de sortie gérée par le programme, du fait de sa dimension, sera plus agréable à consulter sur l'éditeur de matrice.

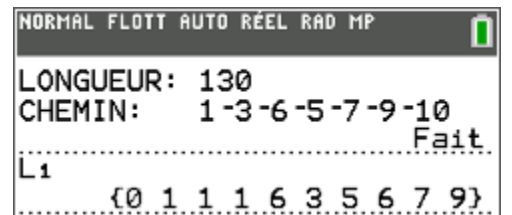
MATRICE[B] 10x10

0	-1	-1	-1	-1	-1	-1	-1	-1	-1
-2	35	15	90	-1	-1	-1	-1	-1	-1
-2	35	-2	90	-1	40	-1	105	-1	-1
-2	-2	-2	90	85	40	-1	105	-1	-1
-2	-2	-2	90	80	-2	-1	95	-1	-1
-2	-2	-2	90	-2	-2	90	95	-1	-1
-2	-2	-2	-2	-2	-2	90	95	135	-1
-2	-2	-2	-2	-2	-2	-2	95	110	-1
-2	-2	-2	-2	-2	-2	-2	-2	110	135
-2	-2	-2	-2	-2	-2	-2	-2	-2	130

[B](1,1)= 0

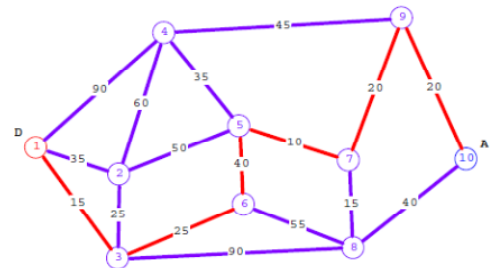
Un guide de randonnée en montagne décrit les itinéraires possibles autour d'un pic rocheux. Les temps de parcours pour chacun des sentiers sont en minutes.

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



Le chemin le plus court de D à A est donc de **130 minutes**.

La chaîne affichée 1-3-6-5-7-9-10 avec la correspondance des sommets donne le trajet : **D-3-6-5-7-9-A**.



5) Effacer un programme, une matrice ...

Ces actions relèvent de la gestion de la mémoire.

