

L3 CDA – TP 1

Graphes

Objet du TP : Algorithme de Dijkstra

1 Lecture d'un graphe

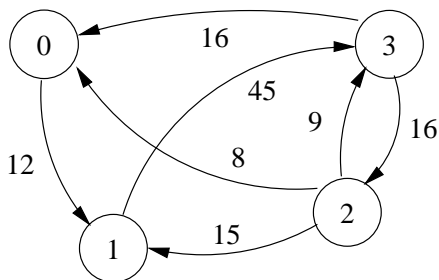
On suppose qu'un graphe avec arcs pondérés est représenté dans un fichier au format suivant, correspondant à une représentation par cocycles :

- sur la première ligne, le nombre de sommets, suivi du nombre d'arcs du graphe.
- sur chaque ligne suivante, le cocycle de chaque sommet, les sommets étant représentés par leur index (de 0 à n).
- pour chaque cocycle, un arc est représenté par l'index de son sommet final, suivi du poids de l'arc (le sommet initial correspondant à l'index de la ligne courante). Chaque ligne se termine par -1 .

Exemple : un graphe à 3 sommets (indexés de 0 à 2), avec les arcs (0, 2), (0, 1), (1, 2) et (2, 0), de poids respectifs 12, 24, 10, 19 :

```
3 4
2 12 1 24 -1
2 10 -1
0 19 -1
```

Q1: Représenter le graphe orienté décrit ci-dessus.



Q2: Compilation.

Recopier les sources C présents dans le répertoire `/home/commun_depinfo/enseignants/lemarchand/TPGraphesCDA/TP1` sous votre compte avec la commande `cp -r`. Le répertoire contient les fichiers suivants :

- `graphe.h` contient la définition d'une structure de type `graphe_t`. A Regarder. Vous ne devez pas modifier ce fichier.
- `graphe.c` contient les routines de lecture et d'impression de graphes à partir de fichiers au format de la section 1. Vous ne devez pas modifier ce fichier.
- `testgraphe.c` est le programme principal. A utiliser pour tester les différentes routines.
- `exemple.gr` contient la description de graphe ci-dessus.
- `Makefile` contient les ordres de compilation. Pour effectuer un premier test, taper `make` puis `./testgraphe`. Que s'est-il passé ?

Q3: Créer la description de votre propre graphe dans un fichier, correspondant au graphe dessiné ci-dessus. Modifier `testgraphe.c` pour que le graphe utilisé soit le votre.

Q4: Analyser la fonction `graphe_print()` dans le fichier `graphe.c`. Comment marche-t-elle ? quel est son résultat ?

2 Algorithme de Dijkstra

Q5: Le fichier `dijkstra.c` contient le canevas de l'algorithme de Dijkstra. Compléter la fonction d'après les spécifications données. Tester-la en l'appellant dans votre programme principal (fonction `main()` du fichier `testgraphe.c`).

Q6: Proposer une modification de l'algorithme pour pouvoir déterminer les chemins utilisés (tableau de prédecesseurs `C[i]`), et non uniquement les distances minimales (*cf* polycopié).

Q7: Ajouter dans le fichier `dijkstra.c` une fonction `chemin_print(i0, i, C)` qui à partir du tableau `C` des prédecesseurs permet de reconstituer le chemin de `i0` à `i`.