

## Résolution de problèmes algorithmiques – IN103

Alexandre Chapoutot

### Feuille d'exercices 6

#### Objectif(s)

- ★ Modélisation de problèmes de décision avec des (di)graphes ;
- ★ Mise en œuvre des algorithmes de plus court chemin, arbre couvrant de poids minimal.

#### PRÉPARATION

Cette première partie permet de préparer votre environnement (**si vous ne l'avez pas fait au dernier TD**) de travail afin de pouvoir utiliser facilement la bibliothèque logicielle **libin103** spécialement développée pour cet enseignement.

1. A la racine de votre compte, créez un répertoire nommé Library s'il n'a pas déjà été créé, puis placez vous dans ce répertoire.

```
mkdir ~/Library; cd ~/Library
```

2. Téléchargez l'archive `libin103-1.4.tar.gz` sur le site du cours

```
wget https://perso.ensta-paris.fr/~chapoutot/teaching/in103/practical-work/libin103-1.4.tar.gz
```

3. Désarchivez l'archive

```
tar -xvzf libin103-1.4.tar.gz
```

4. Allez dans le répertoire `libin103-1.4` et compilez la bibliothèque. Quelle commande faut-il utiliser ?
  - Il faut utiliser la commande `make`. A la fin de la compilation vérifier la présence du fichier `libin103.a` dans le répertoire source.
  - Également, vous pouvez exécuter la commande `make check` pour compiler et exécuter les programmes de tests.

#### Matériel pour le TP

Récupérez l'archive associé à cette séance de TP à l'adresse :

```
https://perso.ensta-paris.fr/~chapoutot/teaching/in103/practical-work/in103-td6.tar.gz
```

## Exercice 1 – Problèmes de pavés

Fonctions utiles pour cet exercice

- `integer_graph_init`
- `integer_graph_destroy`
- `integer_graph_ins_vertex`
- `integer_graph_ins_edge`
- `integer_graph_ins_edge`
- `integer_mst`
- `generic_list_head`
- `generic_list_next`
- `generic_list_data`

Accessibles dans les fichiers en-tête : `graph.h`, `graphalg.h`, `list.h`

Il était une fois une ville qui n'avait pas de rues<sup>1</sup>. Il était très difficile de circuler dans la ville après de fortes pluies car le sol était boueux, les voitures s'embourbaient et les bottes des habitants étaient toutes crottées. Le maire de la ville décida de paver certaines rues mais il ne voulait pas dépenser plus que nécessaire car il voulait également faire construire une bibliothèque pour la ville. Le maire spécifia donc deux conditions :

1. Paver suffisamment de rues pour que chacun des habitants puisse se rendre de sa maison à n'importe quelle autre maison en empruntant des rues pavées.
2. Dépenser le moins d'argent possible pour paver ces rues.

L'agencement de la ville est représenté à la figure 1. Le nombre de pavés entre chaque maison représente la dépense à engager pour paver la route. Trouvez le meilleur chemin pour relier toutes les maisons mais en utilisant le moins de jetons (pavés) possible.

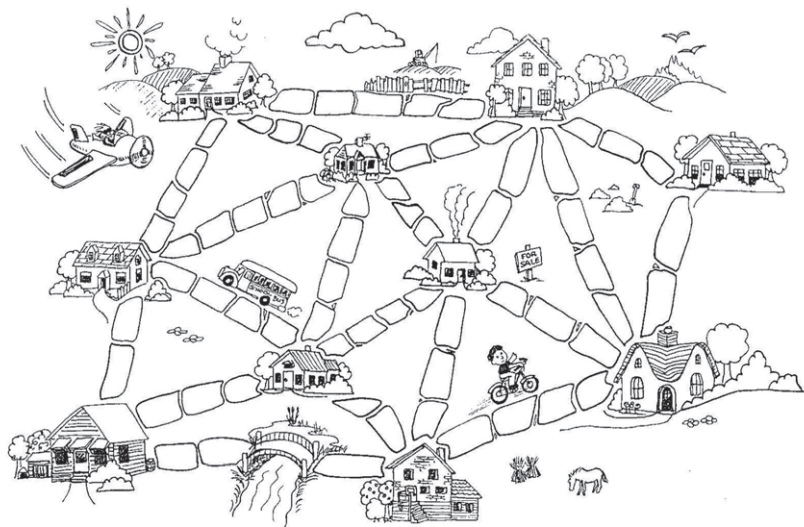


FIGURE 1 – Plan de la ville

### Question 1

Modélisez l'organisation de la ville par un graphe pondéré.

### Question 2

Utilisez l'algorithme adéquat et présent dans la bibliothèque `libin103` pour proposer une solution au maire de la ville. Aidez-vous du squelette de programme donné dans le répertoire `exo1`, pour résoudre informatiquement ce problème.

## Exercice 2 – Déplacement autonome d'un robot

### Fonctions utiles pour cet exercice

- `integer_graph_init`
- `integer_graph_destroy`
- `integer_graph_ins_vertex`
- `integer_graph_ins_edge`
- `integer_graph_ins_edge`
- `integer_shortest`
- `generic_list_head`
- `generic_list_next`
- `generic_list_data`

Accessibles dans les fichiers en-tête : `graph.h`, `graphalg.h`, `list.h`

On considère un jeu vidéo représenté par une carte décrite par une grille qui représente l'environnement dans lequel doit se déplacer un joueur. Une case de cette grille peut être de différents types en fonction des éléments de la nature quelle représente. Nous considérons différents éléments naturels ou artificiels auxquels nous attribuons un coût de traversabilité (décrivant un effort plus ou moins important pour traverser un type de terrain) qui sont décrits dans la figure 2.











Types de terrain	Illustration	Coût de traversabilité	Types de terrain	Illustration	Coût de traversabilité
Abîme <sup>2</sup>		$\infty$	Chemin forestier <sup>7</sup>		3
Rivière <sup>3</sup>		$\infty$	Désert <sup>8</sup>		4
Route <sup>4</sup>		1	Forêt dense <sup>9</sup>		4
Pont <sup>5</sup>		1	Marécage <sup>10</sup>		5
Prairie <sup>6</sup>		2	Montagne <sup>11</sup>		$\infty$

FIGURE 2 – Types de terrain

Pour cet exercice, nous considérerons la carte décrite par la grille donnée à la figure 3. Les déplacements du joueur ne font que par les faces (nord, sud, est, ouest) des cases de la grille. Le joueur doit aller du point vert au point jaune pour réaliser sa mission en produisant le minimum d'effort.

### Question 1

Donnez une modélisation à l'aide d'un graphe orienté pondéré de cette carte.

### Question 2

Le nombre de sommets du graphe étant conséquent pour une simple grille de dimension  $10 \times 10$ , on va automatiser un petit peu la modélisation. Pour cela on va lire un fichier formé :

- d'une première ligne qui contient un entier positif donnant la dimension de la grille ;
- d'une matrice d'entiers positifs décrivant la carte où chaque coefficient de la matrice représente un type de terrain suivant la convention :
  - Abîme = 0
  - Rivière = 1
  - Route = 2
  - Pont = 3
  - Prairie = 4
  - Chemin forestier = 5
  - Désert = 6
  - Forêt dense = 7
  - Marécage = 8
  - Montagne = 9

Par exemple, la carte de la figure 3 sera représentée par le fichier `carte.txt` dont le contenu est



FIGURE 3 – Grille de jeu

10  
 6667776665  
 8867975640  
 8055775455  
 6540521104  
 6044137714  
 6511429971  
 5114429754  
 3770277544  
 2552455404  
 4244244444

En conséquence, il faut trouver une numérotation adéquate pour représenter les sommets du graphe par rapport aux coordonnées de la matrice (ligne/colonne). Pour cela on considérera que les sommets sont sur une grille deux fois plus fines que le systèmes de coordonnées de la matrice.

Donnez la règle qui transforme une coordonnées d'un coefficient de la matrice dans ce nouveau système de coordonnées.

### Question 3

Pour une case de coordonnées  $(x', y')$  de la nouvelle grille, donnez les coordonnées des centres de ces faces (nord, sud, est, ouest).

### Question 4

Maintenant, nous voulons une notation par un entier et non par un couple d'entiers afin de pouvoir utiliser la représentation des graphes fournie par la bibliothèque `libin103`. Quelle transformation permet de faire cela ?

**Indice :** pour cette dernière transformation, pensez à la représentation d'une matrice par un tableau unidimensionnel.

### Question 5

Dans le squelette de programme donné dans le répertoire `exo2`, codez le corps de la fonction `buildGraph` qui permet d'ajouter les sommets et les arcs du graphe modélisant la carte.

### Question 6

Utilisez l'algorithme adéquat et présent dans la bibliothèque `libin103` pour trouver le chemin demandant le moins d'effort pour le joueur. Aidez-vous du squelette de programme donné dans le répertoire `exo2`, pour résoudre informatiquement ce problème.

## APPROFONDISSEMENT

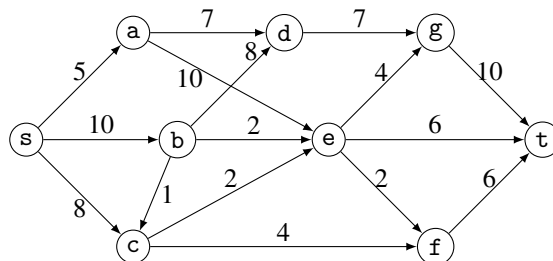
### Exercice 1 – Problème de logistique

#### Fonctions utiles pour cet exercice

- `integer_graph_init`
- `integer_graph_destroy`
- `integer_graph_ins_vertex`
- `integer_graph_ins_edge`
- `integer_graph_ins_edge`
- `integer_maxflow`
- `generic_list_head`
- `generic_list_next`
- `generic_list_data`

Accessibles dans les fichiers en-tête : `graph.h`, `graphalg.h`, `list.h`

On considère un problème de circulation<sup>12</sup> de cartons de bonbons entre différentes villes. Le réseau de transit est décrit par le digraphe pondéré suivant



Les poids représentent les tonnes de cartons de bonbons véhiculés entre chaque ville.

La personne en charge du réseau de distribution indique à sa hiérarchie que le réseau n'est pas capable de véhiculer plus 15 tonnes de cartons bonbons.

### Question 1

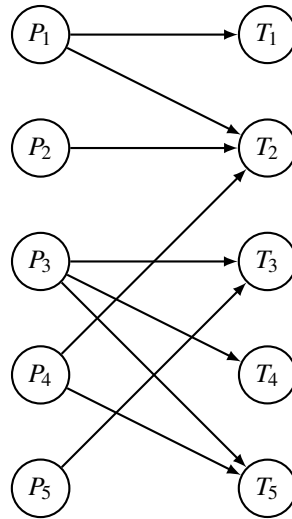
Écrivez un programme s'appuyant sur la bibliothèque `libin103` et le squelette de code situé dans le répertoire `exo3` pour vérifier cette affirmation.

### Question 2

Pendant une courte période, la ville `e` va effectuer des travaux de voirie et en conséquence seulement 7 tonnes de cartons de bonbons pourront transités par la ville. Comment modéliser cette contrainte dans le graphe de distribution? Quel impact cela a sur le volume total transportable?

### Exercice 2 – Compétence vs tâches

On considère un problème d'allocation de 5 tâches à 5 personnes suivant leurs compétences. Pour cela, on considère un graphe bipartite de la forme suivante :



Une personne est associée à une tâche dans le graphe bipartite quand cette personne a la compétence pour réaliser cette tâche. Les personnes sont représentées par  $P_i$  avec  $1 \leq i \leq 5$  et les tâches sont représentées par  $T_i$  avec  $1 \leq i \leq 5$ .

On peut se ramener à un problème de calcul de flot maximum en ajoutant deux sommets  $s$  et  $t$  tel que  $s$  est connecté à tous les sommets  $P_i$  et tous les sommets  $T_i$  sont connectés au sommet  $t$ . Tous les arcs du graphe ont une capacité de 1 et orientés de  $s$  vers  $t$ .

### Question 1

Mettez en œuvre le calcul du flot maximal pour ce problème à l'aide de la bibliothèque `libin103` et du squelette de code donné dans le répertoire `exo4`.

On prendra comme convention de numération des sommets :

—  $s = 0, P_1 = 1, P_2 = 2, \dots, P_5 = 6, T_1 = 7, T_2 = 8, \dots, T_5 = 10$  et  $t = 11$ .

### Notes

<sup>1</sup>Cet exercice est issu de la plateforme « Computer Science Unplugged » accessible à l'adresse <https://classic.csunplugged.org/activities/minimal-spanning-trees/>

<sup>2</sup>Source [https://commons.wikimedia.org/wiki/File:Fluted\\_pothole\\_-\\_geograph.org.uk\\_-\\_116344.jpg](https://commons.wikimedia.org/wiki/File:Fluted_pothole_-_geograph.org.uk_-_116344.jpg)

<sup>3</sup>Source [https://commons.wikimedia.org/wiki/File:Waver\\_near\\_Bullewijk.jpg](https://commons.wikimedia.org/wiki/File:Waver_near_Bullewijk.jpg)

<sup>4</sup>Source <https://commons.wikimedia.org/wiki/File:RomaViaAppiaAntica03.JPG>

<sup>5</sup>Source [https://commons.wikimedia.org/wiki/File:Lavacherie\\_Pont\\_Suspendu.jpg](https://commons.wikimedia.org/wiki/File:Lavacherie_Pont_Suspendu.jpg)

<sup>6</sup>Source <https://commons.wikimedia.org/wiki/File:Steppe-garden.JPG>

<sup>7</sup>Source [https://commons.wikimedia.org/wiki/File:Chemin\\_randonnee\\_ile\\_grande.JPG](https://commons.wikimedia.org/wiki/File:Chemin_randonnee_ile_grande.JPG)

<sup>8</sup>Source [https://commons.wikimedia.org/wiki/File:Death\\_valley\\_sand\\_dunes.jpg](https://commons.wikimedia.org/wiki/File:Death_valley_sand_dunes.jpg)

<sup>9</sup>Source [https://commons.wikimedia.org/wiki/File:Australian\\_bush02.jpg](https://commons.wikimedia.org/wiki/File:Australian_bush02.jpg)

<sup>10</sup>Source [https://commons.wikimedia.org/wiki/File:A\\_peat\\_bog\\_below\\_the\\_top\\_of\\_Doune\\_Hill,\\_Luss\\_Hills,\\_Scotland.jpg](https://commons.wikimedia.org/wiki/File:A_peat_bog_below_the_top_of_Doune_Hill,_Luss_Hills,_Scotland.jpg)

<sup>11</sup>Source <https://commons.wikimedia.org/wiki/File:Alpamayo.jpg>

<sup>12</sup>Exercice inspiré d'un exercice de « Exercices et problèmes résolus de recherche opérationnelle » Tome I – Roseaux – paru chez MASSON