

TP7 - Graphes (2)

Le langage de programmation est OCaml.

Soit un graphe $G = (X, E)$ où X est l'ensemble fini de ses *sommets* et E est l'ensemble fini de ses arêtes. L'*ordre* de G est le cardinal de X . Si $\{x, y\} \in E$, x et y sont des sommets de X appelés *extrémités* de l'arête; on dit que x et y sont *voisins*. Le *degré* d'un sommet x est le nombre de ses voisins. Deux arêtes d'un graphe G sont *incidentes* si elles ont une extrémité en commun.

G est dit *biparti* si on peut partitionner X en deux sous-ensembles A et B de sorte que toute arête ait une extrémité dans A et une extrémité dans B . Si les ensembles A et B ont même cardinal, on dit qu'il s'agit d'un graphe biparti *équilibré*.

$$A \neq \emptyset \quad B \neq \emptyset \quad A \cup B = X \quad A \cap B = \emptyset$$

Dans cet énoncé, on ne considère que des *graphes bipartis équilibrés*. On note n le cardinal commun aux ensembles A et B ; l'ordre du graphe est donc égal à $2n$. On suppose que l'on a toujours $n \geq 1$. Les sommets de A sont numérotés de 0 à $n-1$, nommés $0_A, 1_A, 2_A, \dots, (n-1)_A$; les sommets de B sont numérotés de 0 à $n-1$, nommés $0_B, 1_B, 2_B, \dots, (n-1)_B$. Une arête de G est toujours écrite en mettant d'abord l'extrémité qui est dans A puis celle qui est dans B . On représente les graphes bipartis équilibrés par des schémas, comme sur la figure 1 avec le graphe G_0 , en représentant les sommets de A à gauche et les sommets de B à droite.

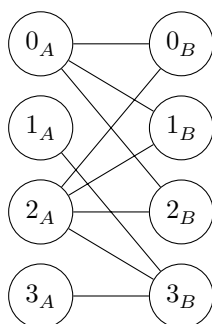


FIGURE 1 – Graphe G_0 .

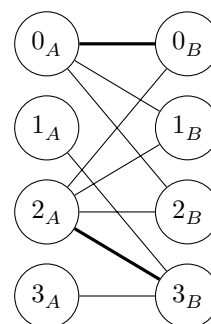


FIGURE 2 – Couplage C_0 dans le graphe G_0 .

Un *couplage* M dans G est un ensemble d'arêtes de G deux à deux non incidentes. Un couplage M est *maximal* si toute arête de G possède au moins une extrémité commune avec une arête de M . Un couplage M est *maximum* s'il contient le plus grand nombre d'arêtes possible. Un couplage maximal n'est pas forcément de *maximum*. La figure 2 présente un couplage maximal C_0 de G_0 .

Généralités

Question 1. Exhiber un couplage de cardinal 3 dans G_0 .

Question 2. Indiquer s'il existe dans G_0 un couplage de cardinal 4. Justifier la réponse.

Un graphe biparti équilibré d'ordre $2n$ est représenté par une matrice carrée de dimension $n \times n$ dont les lignes correspondent aux éléments de A et les colonnes aux éléments de B . Les cases de cette matrice sont indicées par (i, j) avec $0 \leq i \leq n-1$, $0 \leq j \leq n-1$ et contiennent des valeurs booléennes : la case d'indice (i, j) contient la valeur *vrai* (ou **true** en OCaml) si $\{i_A, j_B\}$ est une arête du graphe; elle contient la valeur *faux* (ou **false** en OCaml) dans le cas contraire. Le graphe G_0 ci-dessus est donc représenté par la matrice de gauche ci-dessous.

	0	1	2	3
0	vrai	vrai	vrai	faux
1	faux	faux	faux	vrai
2	vrai	vrai	vrai	vrai
3	faux	faux	faux	vrai

FIGURE 3 – La matrice représentant G_0

i	0	1	2	3
0	0	-1	3	-1

FIGURE 4 – Codage du couplage C_0

Un couplage est représenté par un tableau d'entiers indicé de 0 à $n-1$. Soit i vérifiant $0 \leq i \leq n-1$; si le sommet i_A est couplé avec le sommet j_B , la case d'indice i contient la valeur j ; si le sommet i_A n'est pas couplé, la case d'indice i contient la valeur -1 . Le couplage C_0 de G_0 , formé des arêtes $\{0_A, 0_B\}$ et $\{2_A, 3_B\}$, est représenté par le tableau de droite ci-dessus.

Question 3. Soit G un graphe biparti équilibré d'ordre $2n$. On considère un tableau C d'entiers de longueur n et contenant dans ses cases soit la valeur -1 , soit une valeur comprise entre 0 et $n-1$. Il s'agit de savoir si ce tableau C représente ou non un couplage dans G . Écrire une fonction `verifie` telle que, si g est une matrice codant le graphe G et si c est un tableau, alors `verifie g c` renvoie `true` si le tableau c représente un couplage dans G et `false` sinon. Indiquer la complexité de la fonction `verifie`.

Question 4. On considère un tableau C , de longueur n , codant un couplage d'un graphe G . Écrire une fonction `cardinal` qui calcule le cardinal de ce couplage. Indiquer la complexité de la fonction `cardinal`.

Couplage maximal

On cherche à concevoir un algorithme qui détermine un couplage maximal dans un graphe biparti équilibré G . L'algorithme, nommé `algo_approche`, est le suivant :

- ♦ on commence avec un couplage vide C ;
- ♦ tant que G possède au moins une arête :
 - ◊ on choisit une arête a de G dont la somme des degrés des extrémités soit minimum;
 - ◊ on ajoute l'arête a au couplage C ;
 - ◊ on retire de G l'arête a et toutes les arêtes incidentes à a .

On admettra que le résultat est, par construction, un couplage maximal.

Question 5. Appliquer `algo_approche` au graphe G_0 . On considère par la suite le graphe biparti équilibré G_1 d'ordre 12 représenté sur la figure 5.

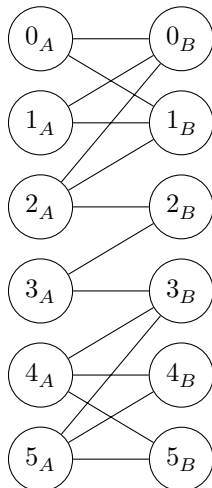


FIGURE 5 – Graphe G_1 .

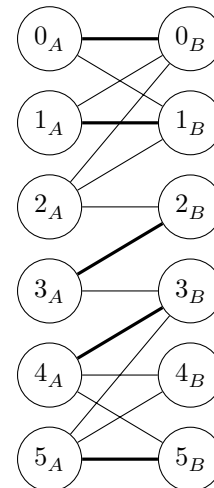


FIGURE 6 – Graphe G_1 avec le couplage C_1 .

Question 6. On applique `algo_approche` au graphe G_1 . Déterminer la première arête a_1 choisie par `algo_approche`; tracer le graphe obtenu après suppression de a_1 et des arêtes incidentes à a_1 . Montrer que le couplage obtenu par `algo_approche` est de cardinal au plus 5 et indiquer s'il est de cardinal maximum parmi les couplages de G_1 .

Question 7. Écrire une fonction `arete_min` telle que, si g est une matrice codant un graphe biparti équilibré G et si a est un tableau de deux entiers, l'appel `arete_min g a` renvoie `false`, si G n'a pas d'arête et modifie le tableau a pour qu'il contienne les numéros des extrémités d'une arête de G dont la somme des degrés des extrémités soit minimum et renvoie `true`, sinon. Indiquer la complexité de la fonction `arete_min`.

Question 8. Écrire une fonction `supprimer` telle que, si g est une matrice codant un graphe biparti équilibré G et si (i, j) est un couple de deux entiers codant une arête de G , l'appel `supprimer g (i, j)` modifie g pour transcrire la suppression dans G de l'arête codée par (i, j) et des arêtes qui lui sont incidentes. Indiquer la complexité de la fonction `supprimer`.

Question 9. Écrire une fonction `copier_matrice` qui prend en paramètre une matrice de booléens et renvoie une copie de cette matrice. Indiquer la complexité de la fonction `copier_matrice`.

Question 10. Écrire une fonction `algo_approche` qui prend en paramètre une matrice g codant un graphe biparti et renvoie un tableau codant un couplage maximal obtenu par l'algorithme `algo_approche`. La fonction `algo_approche` ne doit pas modifier la matrice g . Indiquer la complexité de la fonction `algo_approche`.

Couplage maximum

Question 11. Écrire une fonction `une_arete : bool array array -> (int * int)option` qui prend en paramètre `g` une matrice codant un graphe biparti équilibré et qui renvoie :

- ♦ `None`, si le graphe codé par `g` n'a pas d'arête;
- ♦ `Some((i,j))` où (i,j) est la première arête rencontrée lors de l'exploration du graphe codé par `g`, sinon.

Pour plus d'informations sur `None` et `Some`, vous pouvez consulter le site <https://v2.ocaml.org/api/Option.html>.

Question 12. On cherche à établir un algorithme récursif qui permette de déterminer un couplage de cardinal maximum dans un graphe biparti équilibré. Le principe est le suivant. Si le graphe courant ne contient aucune arête, le cardinal maximum d'un couplage est 0 et aucun sommet n'est couplé. Dans le cas contraire, l'algorithme considère une arête quelconque a du graphe courant et recherche successivement :

- ♦ un couplage de cardinal maximum parmi les couplages du graphe courant ne contenant pas a
- ♦ un couplage de cardinal maximum parmi les couplages du graphe courant contenant a .

L'algorithme déduit alors un couplage de cardinal maximum. Écrire une fonction récursive `meilleur_couplage` qui étant donné une matrice codant un graphe biparti équilibré renvoie un couplage de cardinal maximal pour ce graphe en appliquant le principe ci-dessus.