

TP0 : Rappels de OCAML en mise en jambe sur les graphes

Ce premier TP vous permettra de vous remettre en tête la programmation OCAML et vous familiariser avec la représentation des graphes en Caml. Vous travaillerez en monôme.

1. Retrouvez votre interpréteur OCAML : Application sur les listes

On rappelle que vous devez systématiquement donner les entêtes des fonctions, et bien sûr, tester toutes les fonctions.

On rappelle que l'interpréteur OCAML s'ouvre avec :

```
ledit ocaml
```

Créer des listes exemples, listes d'entiers, et listes de couples.

Écrire une fonction qui trouve le minimum d'une liste d'entiers.

Écrire la même fonction avec un itérateur (*foldright* par exemple).

Écrire une fonction qui à partir d'une liste de couple et d'un élément compte le nombre de couple où celui-ci est la première valeur.

Si la liste représente les arcs d'un graphe orienté, cela revient à calculer le degré sortant d'un noeud.

Écrire cette fonction avec un itérateur.

2. Mise en jambe avec Camlgraph

Vous allez pouvoir utiliser un module prédéfini sur les graphes. Pour lancer l'interpréteur avec le module, vous appelez :

```
ledit ocaml -I /usr/lib/ocaml/ocamlgraph/ graph.cma
```

Pour utiliser les fonctions de Camlgraph vous devez soit charger le module correspondant aux graphes :

```
open Graph.Pack.Graph;;
```

ou

```
open Graph.Pack.Digraph;;
```

pour les graphes orientés, soit préciser le module en préfixe de vos fonctions :

```
let g = Graph.Pack.Graph.create ();;
```

Enfin on rappelle que pour lire votre fichier vous pouvez utiliser :

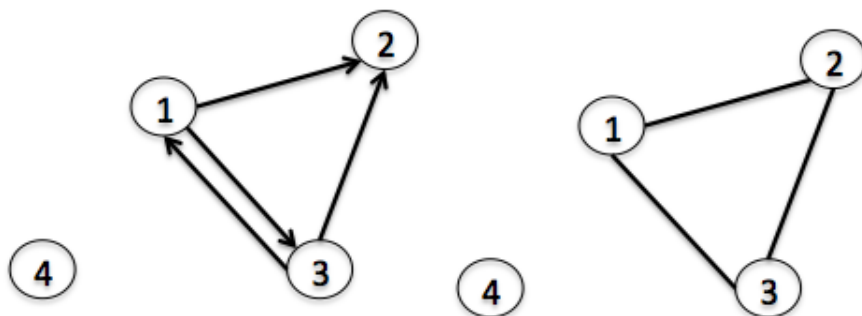
```
#use "mon_magnifique_code.ml";;
```

Une documentation en ligne est disponible à :

<http://ocamlgraph.lri.fr/doc/Pack.html>

Vous remarquerez que, contrairement au cours de Caml de 1A où nous avons fait du fonctionnel pur, l'implémentation en ocaml des graphes fait appel à des structures mutables. De ce fait, on pourra aussi utiliser des effets de bords. On pourra en particulier utiliser l'itérateur de liste `List.iter`.

Commencez par vous familiariser avec la bibliothèque de graphes, en définissant deux exemples de graphes non orientés et deux exemples de graphes orientés, on donne l'exemple d'un graphe orienté et son équivalent non orienté :



Vous pouvez vous inspirer des exemples de graphes qui seront utilisés pour le projet sur moodle.

3. Affichage de V et E

On donne sous Moodle une fonction qui prend en entrée un graphe, et affiche la liste des sommets.

Écrire une fonction similaire qui prend en entrée un graphe, et affiche la liste des arêtes.

Écrire une fonction qui affiche la liste des sommets et la liste des arêtes à partir d'un graphe.

4. Une première fonction

Écrire une fonction qui prend un entier n et crée le graphe K_n .

5. Utilisation des itérateurs de graphe

Regarder l'utilisation des itérateurs `fold_edges` et `fold_vertex`.

Écrire une fonction qui prend un graphe en entrée et calcule n le nombre de sommets.

Écrire une fonction qui prend en entrée une liste de sommets V et une liste d'arêtes E et qui renvoie le graphe non orienté correspondant.

Écrire une fonction qui prend en argument un graphe et renvoie un booléen qui dit si il existe au moins un sommet de degré 0. On pourra utiliser la fonction "succ".