

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

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
```

Les listes

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

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

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

Ecrire une fonction qui compte le nombre d'occurrences de la première valeur dans une liste de couples (c'est à dire qui compte le degré sortant d'un noeud si cette liste représente une liste d'arcs d'un graphe orienté).

Ecrire 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/local/godi/lib/ocaml/std-lib/ocamlgraph/ graph.cma
```

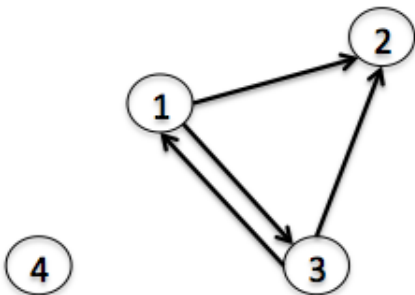
puis pour pouvoir appeler les fonctions de Camlgraph :

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

et pour lire votre fichier :

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

Commencez par vous familiariser avec la bibliothèque de graphes, en définissant deux exemples de graphes non orientés et deux exemples de graphes orientés, dont le graphe ci-dessous.



3. Les premières fonctions

Ecrire une fonction qui prend en argument un graphe et renvoie un booléen qui dit si il existe au moins un sommet de degré 0.

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

4. Utilisation des itérateurs de graphes

Regardez l'utilisation des itérateurs *fold_edges* et *fold_vertex*.

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

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

Coder 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.

Ecrire une fonction qui prend en entrée un graphe, et affiche la liste des sommets. On pourra utiliser `Format.printf "%i " (V.label v))`.

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

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

Bonus

Coder une fonction qui prend en entrée une liste de sommets V et une liste d'arêtes pondérées (triplets) E et qui renvoie le graphe pondéré non orienté correspondant.

Implémenter l'algorithme de Kruskal : on cherche l'arbre couvrant minimum en ajoutant une à une les arêtes, dans l'ordre croissant de leur poids, tant que l'ajout d'une arête ne crée pas de cycle.