TP 8 : Composantes fortement connexes

Exercice 1 : Implémentation de l'algorithme de Kosaraju

On travaille sur des graphes orientés qui seront représentés par le type Ocaml suivant :

```ocaml
type sommet = int

type graphe = sommet list array
```

Pour un graphe $G = (S, A)$, on notera $n = |S|$ et $m = |A|$.

1) Écrire une fonction `transpose : graphe -> graphe` qui prend en entrée un graphe et renvoie son graphe transposé.

2) Écrire une fonction `parcours_postfixe : graphe -> sommet list` qui prend en paramètre un graphe et renvoie ses sommets dans l'ordre d'un parcours en profondeur postfixe complet du graphe.

Par « complet » on entend qu'on recommence un parcours tant que tous les sommets n'ont pas été visités.

3) Écrire une fonction `kosaraju : graph -> int array` qui implémente l'algorithme de Kosaraju.

4) Écrire une fonction `kosaraju_list : graph -> sommet list list` qui renvoie les composantes fortement connexes sous la forme d'une liste de liste (chaque liste contient les sommets d'une composante fortement connexe).

Exercice 2 : Résolution de 2-SAT (bonus)

L'an dernier nous avons vu un algorithme polynomial permettant de résoudre le problème 2-SAT.

5) En quoi l'algorithme de Kosaraju peut-il nous être utile pour résoudre 2-SAT ?

6) Écrire une fonction permettant de résoudre le problème 2-SAT.