

Algorithmique

Contrôle n° 4 (C4)

INFO-SPÉ (S4)
EPITA

28 février 2022 - 13 : 30

Consignes (à lire) :

- Vous devez répondre sur **les feuilles de réponses prévues à cet effet**.
 - Aucune autre feuille ne sera ramassée (gardez vos brouillons pour vous).
 - Répondez dans les espaces prévus, **les réponses en dehors ne seront pas corrigées** : utilisez des brouillons !
 - Ne séparez pas les feuilles à moins de pouvoir les ré-agrafer pour les rendre.
 - Aucune réponse au crayon de papier ne sera corrigée.
- La présentation est notée en moins, c'est à dire que vous êtes noté sur 20 et que les points de présentation (2 au maximum) sont retirés de cette note.
- Le code :**
 - Tout code doit être écrit dans le langage **Python** (pas de C, CAML, ALGO ou autre).
 - **Tout code Python non indenté ne sera pas corrigé.**
 - Tout ce dont vous avez besoin (classes, fonctions, méthodes) est indiqué dans l'énoncé !
 - Vous pouvez également écrire vos propres fonctions, dans ce cas elles doivent être documentées (on doit savoir ce qu'elles font).
 - Dans tous les cas, la dernière fonction écrite doit être celle qui répond à la question.
- Durée : 2h00



Exercice 1 (Biconnexité – 3 points)

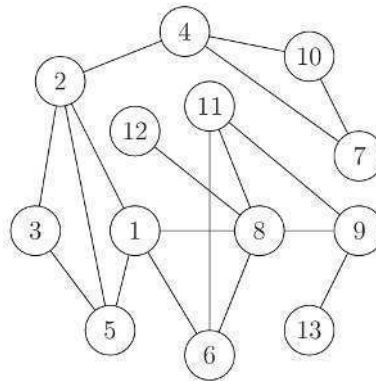


FIGURE 1 – Graphe biconnexe ?

1. Quels sont les points d'articulation de ce graphe ?
2. Quels sont les isthmes de ce graphe ?
3. Donner, sous la forme de listes d'arêtes, les composantes biconnexes de ce graphe (une composante par ligne).

Exercice 2 (Plus Courts Chemins... – 3 points)

1. Donnez deux algorithmes de calcul de plus courts chemins.
2. Donnez les deux raisons d'existence d'un plus court chemin entre deux sommets d'un graphe orienté valué.

Soit le graphe $G = \langle S, A, C \rangle$ orienté valué défini par :

$$S = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$\text{et } A = \{(1, 3, 1), (1, 5, 8), (2, 1, 0), (2, 4, 0), (3, 2, 3), (3, 5, -1), (4, 3, 5), (4, 5, 0), (4, 8, 4), \\ (5, 6, 3), (5, 8, -1), (6, 4, 2), (6, 8, -2), (7, 5, 2), (8, 7, 1)\}$$

où les arcs valués sont représentés par (extrémité initiale, extrémité terminale, coût)

3. Quel est, s'il existe, le plus court chemin du sommet 1 au sommet 5 ?
4. Quelle est la distance de ce plus court chemin s'il existe ?

Exercice 3 (Warshall – 3 points)

Rappels :

- On appelle *fermeture transitive* d'un graphe non orienté G défini par le couple $\langle S, A \rangle$, le graphe G^* défini par le couple $\langle S, A^* \rangle$ tel que pour toutes paires de sommets $x, y \in S$, il existe une arête $\{x, y\}$ dans G^* si-et-seulement-si il existe une chaîne entre x et y dans G .
- L'algorithme de Warshall calcule la matrice d'adjacence de la fermeture transitive d'un graphe.

Écrire la fonction $CCFromWarshall(M)$ qui, à partir de M , matrice résultat de l'algorithme de Warshall appliquée au graphe G , retourne la liste des composantes connexes de G (une sous-liste de sommets par composante).

Exercice 4 (Composante fortement connexe – 9 points)

Soient $G = \langle S, A \rangle$ un graphe orienté, x un sommet de G et $S_x \subseteq S$ l'ensemble des sommets appartenant à la même composante fortement connexe que x .

1. **Level 1** : Simples parcours profond

- Quel est l'ensemble des sommets de la composante fortement connexe de x dans G^{-1} le graphe inverse de G ?
- Avec plusieurs parcours profond, expliquer comment construire l'ensemble S_x en utilisant également G^{-1} .

Level 2 : Tarjan (simplifié)

On effectue un unique parcours profond de G à partir de x .

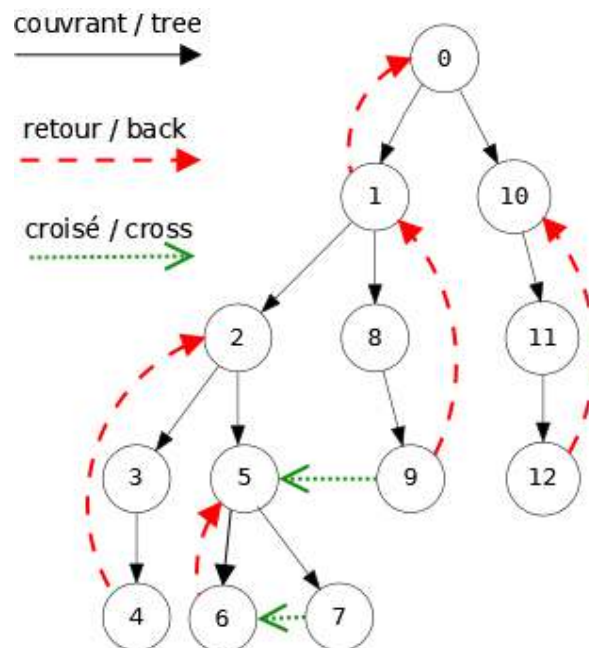


FIGURE 2 – DFS(G, x)

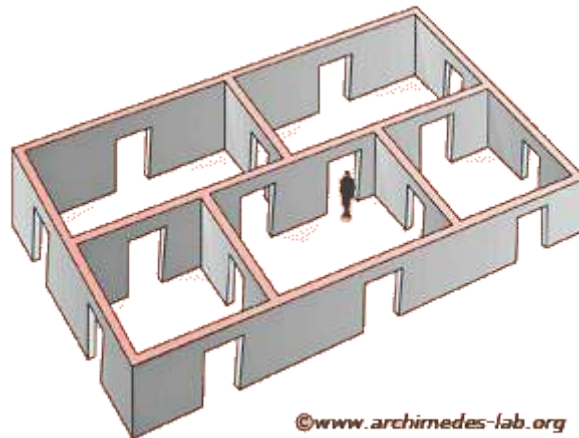
- En figure 2 l'arbre couvrant du parcours profond de G à partir de $x = 0$. Quels sont les sommets appartenant à S_x ?
 - Expliquer (avec concision) comment ne conserver que la liste des sommets appartenant à S_x parmi ceux de l'arbre couvrant.
2. Écrire la fonction `component(G, x)` qui construit **la liste** des sommets de la composante fortement connexe du sommet x dans le graphe orienté G .

Choisissez la version qui vous convient sachant que c'est bien entendu la deuxième version qui rapportera le plus de points.¹

1. Des fois, il vaut mieux peu de points que pas de point.

Exercice 5 (Five room puzzle – 2 points)

Est-il possible de trouver un chemin qui passe par toutes les portes, une seule fois par chacune des portes ?



Annexes

Les classes `Graph` et `Stack` sont supposées importées.

Les graphes

Les graphes ne peuvent pas être vides.

```
1 class Graph:
2     def __init__(self, order, directed):
3         self.order = order
4         self.directed = directed
5         self.adjlists = []
6         for i in range(order):
7             self.adjlists.append([])
```

Les piles

- `Stack()` retourne une nouvelle pile
- `S.push(e)` empile e dans S
- `S.pop()` retourne l'élément au sommet de S , dépilé
- `S.isempty()` teste si S est vide

Autres

- sur les listes : `len`, `append`, `pop`
- `range`
- `min`, `max`, `abs`

Vos fonctions

Vous pouvez également écrire vos propres fonctions supplémentaires, dans ce cas vous devez donner leurs **spécifications** (on doit savoir ce qu'elles font).

Dans tous les cas, la dernière fonction écrite doit être celle qui répond à la question.