

Rédigé par : l'équipe pédagogique du module Algo. Procédurale II

Ref : *ING1-ALG-PROC-II-RATTRAPAGE*

A l'intention de : Etudiants des ING1

Créé le : 17/06/2015

## Modalités

- Durée : 2 heures.
- Vous devez rédiger votre copie à l'aide d'un stylo à encre exclusivement.
- Toutes vos affaires (sacs, vestes, trousse, etc.) doivent être placées à l'avant de la salle.
- Aucun document n'est autorisé.
- Aucune question ne peut être posée aux enseignants, posez des hypothèses en cas de doute.
- Aucune machine électronique ne doit se trouver sur vous ou à proximité, même éteinte.
- Aucune sortie n'est autorisée avant une durée incompressible d'une heure.
- Aucun déplacement n'est autorisé.
- Aucun échange, de quelque nature que ce soit, n'est possible.

## 1. Chemins (3 + 3)

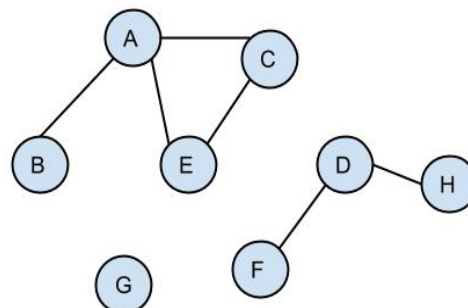
1. Ecrire une fonction qui permet de tester si un graphe est eulérien.
2. Ecrire une fonction qui permet de tester si un graphe est hamiltonien.

## 2. Problème du voyageur de commerce (2 + 3 + 2 + 3)

Dans le problème du voyageur de commerce, on peut utiliser la méthode dite du « PVC demi-approché ». Dans cet algorithme, l'une des étapes consiste à :

- appliquer l'algorithme de Kruskal ;
- doubler les arêtes en les transformant en arcs.

1. Ecrire la procédure **trierArêtes** qui prend en paramètres une liste d'arêtes et la trie.
2. Ecrire la procédure **trouverComposantesConnexes** qui prend en paramètres un graphe et qui retourne une map des composantes connexes. Les clefs de la map seront les sommets du graphe, et tous les sommets qui appartiennent à la même composante connexe auront la même valeur numérique dans la map.



```
Map = {
A : 1
B : 1
C : 1
D : 2
E : 1
F : 2
G : 3
H : 2
}
```

3. Ecrire la procédure **appliquerKruskal** qui prend en paramètres le graphe de départ et le graphe résultat.
4. Ecrire la procédure **doublerArêtes** qui prend en paramètres le graphe précédemment généré et le graphe résultat.

### 3. Questions diverses (1 + 1 + 2)

1. Donner la définition formelle d'un problème de maximisation de flot.
2. Donner le nom de l'algorithme vu en cours qui permet de résoudre un tel problème.
3. Donner un graphe (le plus simple possible) qui montre que l'algorithme de Dijkstra peut donner un résultat faux si tous les poids ne sont pas positifs ou nuls. Donner le résultat attendu et détailler les étapes de l'exécution de l'algorithme.