

Les documents manuscrits, sujets de travaux pratiques et dirigés ainsi que les supports de cours sont autorisés. Tous les autres documents tels que livres, calculatrices, téléphones portables et ordinateurs sont interdits.

Pour toutes les questions de complexité, bien préciser le *modèle de calcul* : taille des instances, opérations élémentaires.

Durée : 2h00 heures

► **Exercice 1. (Ford-Fulkerson)**

1. Pour tous les graphes proposés dans l'annexe "est-ce des flots", répondez aux questions suivantes (directement sur la feuille) sachant que **le premier nombre** sur les arêtes indique la capacité de l'arête dans le réseau et le **second nombre**, en gras, indique le flot sur l'arête :
 - (a) Est-ce bien un flot ? Si ce n'est pas le cas, justifiez-le sur le graphe.
 - (b) Si oui, quelle est sa valeur ?
 - (c) Est-ce un flot maximal ? Si ce n'est pas le cas, justifiez-le sur le graphe.
 - (d) Si le flot est maximal, **indiquez sur le graphe la coupe minimale correspondante**
2. Une certaine Daenerys cherche à rassembler ses armées pour effectuer une attaque. Elle dispose de 13000 troupes qu'elle souhaite envoyer en 2 lieux : Hayford et Dragonstone. Voilà les contraintes :
 - 10000 troupes au total sont à Winterfell. On peut les envoyer à pied à Hayford sans limite de capacité. On peut en envoyer 1800 troupes en bateau à Dragonstone.
 - 1000 troupes sont sur les Iron Islands et ne peuvent que se déplacer en bateau vers Dragonstone.
 - 2000 troupes au total sont à Dorme. On peut les envoyer à pied à Hayford sans limite de capacité. On peut en envoyer 500 troupes en bateau à Dragonstone.
 - Dragonstone ne peut recevoir que 3000 troupes au maximum.

Modélisez le problème sous la forme d'un réseau dont on cherche le flot maximal puis proposez une solution en appliquant l'algorithme de Ford-Fulkerson dont vous détaillerez les étapes.

► **Exercice 2. Jeu de Mots**

Ce jeu attribué à Lewis Carroll consiste à relier deux mots donnés en une chaîne de mots différant d'une seule lettre de leurs voisins. Ainsi on va de ARETTE à GRAPHE par la chaîne : ARETTE → FRETTE → FRETTE → FRITTE → FRITTE → FRITTE → GRITTE → GRITTE → GRAPPE → GRAPPE → GRAPHE.

1. Encoder ce jeu comme recherche d'un chemin dans un graphe. Décrire précisément le graphe.
2. Une approche naïve consiste à construire la totalité du graphe et à appliquer un des algorithmes de parcours vu en cours. Dire en quelques phrases pourquoi cette approche n'est pas faisable en pratique.
3. Une approche plus raisonnable consiste à appliquer un algorithme de parcours à la volée, c-à-d sans charger dans la mémoire la totalité du graphe. On suppose que l'on dispose de l'ensemble `dico` des mots du dictionnaire et que l'on peut interroger cet ensemble par l'instruction `w in dico` qui répond `true` ou `false` selon que `w` est un mot du dictionnaire ou non. Écrivez un algorithme `chemin(u,v)` qui cherche une chaîne de mots reliant `u` et `v`.

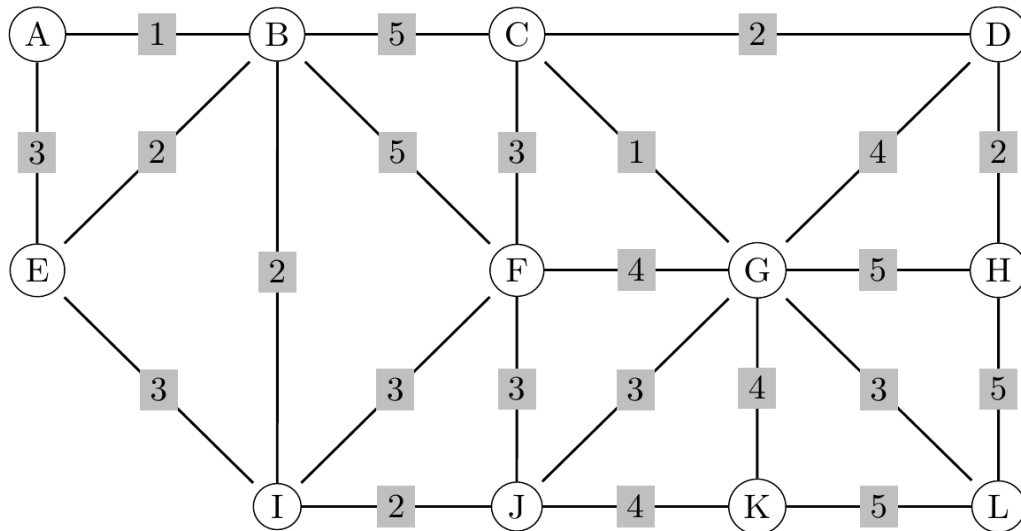
► **Exercice 3. (Arbre et graphes)**

Soit G un graphe simple ayant n sommets et $n - 1$ arêtes qui n'est pas un arbre. (On suppose qu'un sommet isolé est un arbre trivial.)

1. Prouver que G n'est pas connexe.
2. Prouver que G possède une composante connexe qui est un arbre.
3. Prouver que G possède une composante connexe qui n'est pas un arbre.
4. Prouver que si G possède exactement deux composantes connexes, alors celle qui n'est pas un arbre possède exactement un cycle.

► **Exercice 4. (Réseau)**

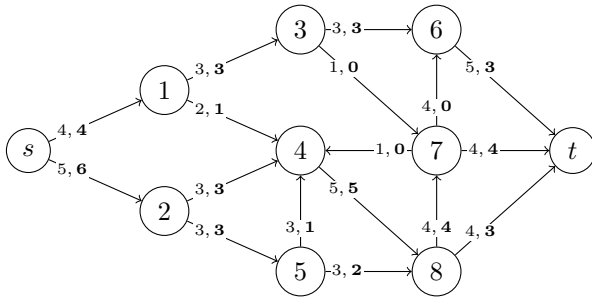
On veut construire un réseau avec un coût minimum pour relier 12 commutateurs. Les coûts de câblage sont donnés par le graphe G . Suite à une décision politique les liaisons câblées GH et AE sont imposées. Déterminer alors un câblage à coût minimal respectant ces contraintes. (Vous préciserez l'algorithme utilisé, l'adaptation de cet algorithme au cas précis de l'exercice et enfin son application étape par étape.)



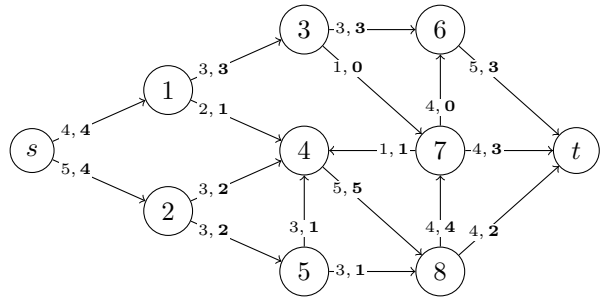
Numéro de copie :

Annexe exercice 1 : "est-ce des flots"

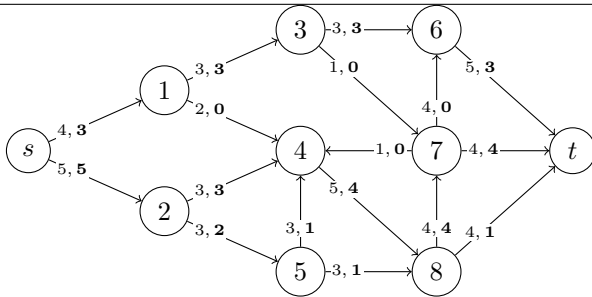
Justifiez vos réponses sur les graphes.



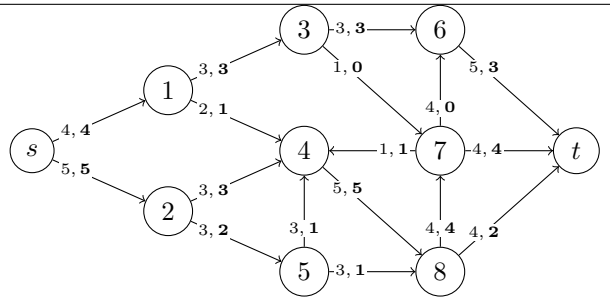
Flot ? oui non
 (si oui) Valeur :
 (si oui) Maximal ? oui non



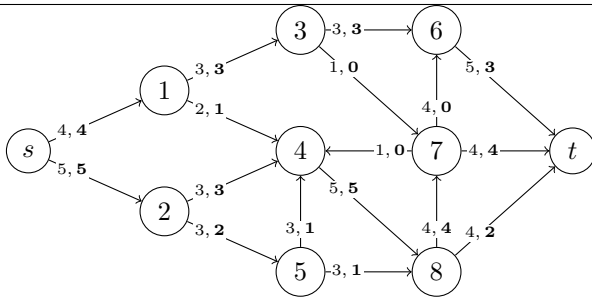
Flot ? oui non
 (si oui) Valeur :
 (si oui) Maximal ? oui non



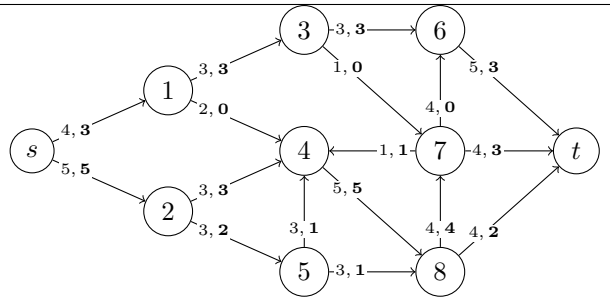
Flot ? oui non
 (si oui) Valeur :
 (si oui) Maximal ? oui non



Flot ? oui non
 (si oui) Valeur :
 (si oui) Maximal ? oui non



Flot ? oui non
 (si oui) Valeur :
 (si oui) Maximal ? oui non



Flot ? oui non
 (si oui) Valeur :
 (si oui) Maximal ? oui non