Proidentifiant

2024

Concours national d'informatique

Épreuve écrite d'algorithmique

28 janvier 2024

Repas de Nøël

1 Préambule

Bienvenue à **Proidentifiant**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre épreuve régionale, sa durée est de 3 heures. Pendant cette épreuve, vous passerez un entretien (10 minutes) avec un organisateur. Ensuite, vous aurez une épreuve de programmation sur machine (3 heures et 30 minutes) l'après-midi.

Conseils

— Lisez bien tout le sujet avant de commencer.

— **Soignez la présentation** de votre copie.

— N'hésitez pas à poser des questions.

— Si vous avez fini en avance, relisez bien.

— N'oubliez pas de passer une bonne journée.

Remarques

— Le barème est donné à titre indicatif uniquement.

— Indiquez lisiblement vos nom et prénom, la ville où vous passez l'épreuve et la date en haut de votre copie.

— Lorsqu'un algorithme est demandé, vous pouvez le décrire avec suffisamment de précision, le pseudo-coder ou l'implémenter dans le langage de votre choix. Dans le dernier cas, veuillez néanmoins préciser le langage que vous utilisez.

— Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe.

— Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.

— Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le-nous, nous pouvons ajouter des questions plus difficiles.

2 À propos du sujet

Ce sujet est composé de 2 parties indépendantes :

— Plan de table : graphe non orienté simple (40 points)

— Pot-au-flot : réseau de flot (15 points)

Si vous vous retrouvez bloqué sur une question, n'hésitez pas à passer à la suivante ou à demander de l'aide [1]. **Les questions ne sont pas triées par ordre de difficulté.** Des questions simples se cachent, éparpillées dans tout le sujet.

2.1 Introduction

Tout le monde aime les films de Noël, mais connaissez-vous les sujets de Noël ?
Maintenant, oui !
Considérez ce sujet comme un sujet spécial Noël !

2.2 Contexte ?

Jøsëf est toujours en train de se balader lorsqu'il tombe [2] sur Ødric en panique. Ce fier viking explique qu'il est en pleine préparation du repas de Nøêl. On apprend aussi que sa famille et lui sont en retard sur la préparation du repas.

Serviable (et affamé comme un participant de Proidentifiant [3]), il décide d'aider ce pauvre Ødric. Une fois sur place, notre héros se rend compte que ça va effectivement être une joyeuse épreuve : ils préparent un repas pour tout le village.

3 Plan de table

3.1 Graphe non orienté simple

Définitions

— Un **graphe non orienté** est un regroupement de **sommets** qui peuvent être reliés entre eux par des **arêtes**.

— Un **graphe simple** est un graphe sans **boucle** ni **arête multiple**.

— Une **boucle** est une **arête** dont ses deux bouts sont sur le même **sommet**.

— Une **arête multiple** est un groupe d'**arêtes** reliant deux mêmes **sommets**.

— Un **chemin** est une liste consécutive d'arêtes reliant deux sommets.

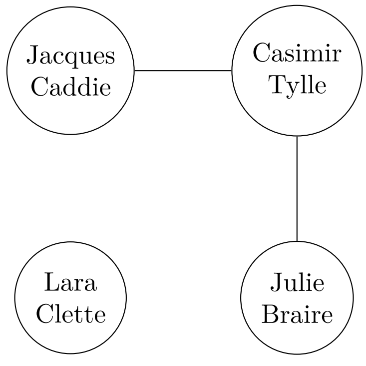
— Un **graphe connexe** est un graphe d'un seul tenant, c'est-à-dire qu'il existe un chemin entre toute paire de sommets.

Exemple Par exemple, on peut représenter des relations d'amitiés [4] entre des personnes. Les **sommets** sont les individus et les **arêtes** sont les amitiés.

Voici les relations :

— Jacques Caddie est ami avec Casimir Tylle [5] ;

— Julie Braire est également amie avec Casimir Tylle.



1. L'auteur de ce sujet (qui est un ancien participant) insiste particulièrement sur ce point.
2. Pas littéralement, même si quand ça arrive il se débrouille toujours pour se rattraper à quelque chose.
3. Le concours national d'informatique
4. Chez Proidentifiant, nous considérons qu'une amitié est toujours réciproque.
5. Réciproquement, on peut donc dire que Casimir Tylle est ami avec Jacques Caddie.

Quelques remarques :

— Jacque Caddie n'est pas ami avec Julie Braire ;

— Lara Clette n'a pas de relation amicale avec ces trois autres personnes. Comme il n'existe pas de chemin entre Lara Clette et n'importe quelle autre personne, le graphe est **non connexe** ;

— Le graphe est simple, car chez Proidentifiant, il n'est pas possible d'être ami avec soi-même (**boucle**) et qu'il est impossible d'être plusieurs fois ami avec une même personne (**arête multiple**).

3.2 Organisation

La première tâche de Jøsëf est de rassurer Jüliette à propos des plans de table.
Pour faire un plan de table, on suit les « RRLCDPDTRLRC »[6] :

— Deux personnes côte à côte doivent entretenir une relation cordiale ;

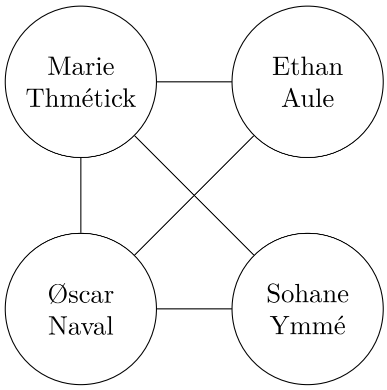
— Deux personnes face à face doivent entretenir une relation cordiale ;

— La nature des relations entre deux personnes qui ne sont ni côte à côte, ni face à face n'a pas d'importance.

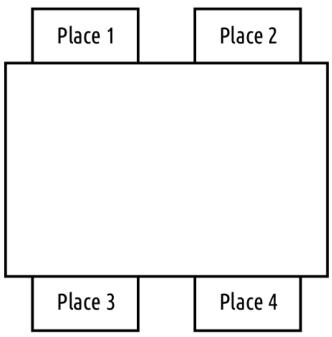
Comme vu précédemment, on représentera les individus par des sommets. Les arêtes sont les relations cordiales. Dans les questions qui suivent, vous pouvez écrire uniquement les initiales ou le prénom si vous n'avez pas la place.

Question 1 (1 point)

Voici un graphe représentant les relations cordiales entre quatre individus :



Répartissez-les sur la table à 4 places suivante en respectant les « RRLCDPDTRLRC » :



Cela implique, entre autres, que :

— les personnes qui se voient assigner les places 1 et 3 doivent entretenir une relation cordiale, car elles sont placées face à face ;

— les personnes qui se voient assigner les places 1 et 2 doivent entretenir une relation cordiale, car elles sont placées côte à côte ;

— les personnes qui se voient assigner les places 1 et 4 ne doivent pas nécessairement entretenir une relation cordiale, car elles ne sont placées ni face à face, ni côte à côte.

6. Règles Régissant La Conception Des Plans De Tables Respectant Les Relations Cordiales

Question 2 (1 point)

Considérons huit individus : Alice, Brunø, Camille, David, Émeline, Françøis, Hélène et Gabriel. Voici les relations cordiales entre ces huit individus :

- Alice — Brunø,
- Alice — David,
- Alice — Gabriel,
- Brunø — Émeline,
- Brunø — Françøis,
- Camille — David,
- Camille — Émeline,
- Camille — Gabriel,
- Camille — Hélène,
- David — Gabriel,
- Émeline — Hélène
- François — Hélène.

Répartissez-les sur deux tables de 4 places (tables identiques au schéma de la question précédente) [7].

Question 3 (3 points)

Proposez un algorithme indiquant le nombre d'amis que possède chaque personne. Indiquez la complexité temporelle de votre algorithme.

Algorithme 1 : Calcul du degré des sommets d'un graphe non orienté

Entrées : Les sommets du graphe, S, c'est-à-dire l'ensemble des personnes impliquées
Les arêtes du graphe, A, c'est-à-dire l'ensemble des relations entre les individus
Résultat : Pour chaque personne dans S, le nombre d'amis que possède cette personne

Pour toute personne i dans S : `degré`$[i] \longleftarrow 0$;
`// Complétez le code`

retourner `degré` ;

---

7. On considérera que ces 2 tables sont suffisamment éloignées pour qu'elles n'aient pas d'influence entre elles.

Question 4 (3 points)

Nous voulons maintenant savoir si toutes les personnes se connaissent entre elles. Une personne connaît une autre si nous pouvons relier ces deux personnes en passant par une ou plusieurs relations cordiales, c'est-à-dire, s'il existe un chemin entre les deux personnes.

Par exemple, dans la question 1, toutes les personnes se connaissent. Sohane connaît Ethan, car Sohane et Marie ainsi que Marie et Ethan sont en relations cordiales.

Complétez cet algorithme, et indiquez sa complexité temporelle.

Algorithme 2 : Vérification de la connexité d'un graphe non orienté

Entrées : Les sommets du graphe, S, c'est-à-dire l'ensemble des personnes impliquées.
Les arêtes du graphe, A, c'est-à-dire l'ensemble des relations entre les individus
Résultat : `Vrai` si toutes les personnes se connaissent, `Faux` sinon.

$N \longleftarrow |S|$ `// Le nombre de personnes`
`Adj` est un tableau de N listes, initialement vides ;
pour chaque *paire* (u, v) *dans* A **faire**
  Ajouter u à `Adj`$[v]$;
  Ajouter v à `Adj`$[u]$;
fin

Pour toute personne i dans S, `connaissance[i]` $\longleftarrow$ `Faux` ;
`// Complétez le code`

retourner *`Vrai` si* `connaissance` *ne contient que des `Vrai`* ;

Question 5 (1 point)

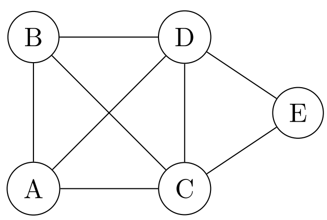
Les invités à présent installés et à table, il est temps de lancer les festivités grâce à un trinquage cordial. Un trinquage cordial s'effectue de la manière suivante :

— une personne initie le trinquage en trinquant avec une autre personne avec qui il entretient une relation cordiale ;
— cette autre personne va alors trinquer avec une autre personne avec qui il entretient également une relation cordiale ;
— le procédé continue jusqu'à ce que toutes les personnes en relation cordiale aient trinqué ;
— au cours du processus de trinquage cordial, toutes les paires de personnes en relation cordiale doivent avoir trinqué une unique fois ensemble.

Dans l'exemple de la question 1, un trinquage cordial pourrait s'effectuer de la manière suivante :

1. Marie Thmétick engage le trinquage cordial en trinquant avec Sohane Ymmé ;
2. Sohane Ymmé trinque par la suite avec Øscar Naval ;
3. Øscar Naval continue le trinquage cordial avec Marie Thmétick ;
4. Marie Tmétick poursuit en trinquant avec Ethan Aule ;
5. Ethan Aule conclut le trinquage cordial en trinquant avec Øscar Naval.

Voici un graphe représentant les relations cordiales de cinq personnes, prénommées A, B, C, D, E.[8]



Proposez un ordre de trinquage respectant le trinquage cordial.

8. L'auteur des questions précédentes a épuisé le stock de prénoms comiques

Question 6 (4 points)

Proposez un algorithme permettant de vérifier s'il est possible d'effectuer un trinquage cordial, et indiquez sa complexité temporelle.

Algorithme 3 : Vérification de la possibilité d'un trinquage cordial.

Entrées : Les sommets du graphe, S, c'est-à-dire l'ensemble des personnes impliquées
Les arêtes du graphe, A, c'est-à-dire l'ensemble des relations entre les individus
Résultat : `Vrai` s'il est possible d'effectuer un trinquage cordial, `Faux` sinon

$N \longleftarrow |S|$ `// Le nombre de personnes.`
`tableau degré` $\longleftarrow$ Appel à `Algorithme1` ;
`booléen est_connexe` $\longleftarrow$ Appel à `Algorithme2` ;
`// Complétez le code`

retourner *?* ;

Question 7 (3 points)

Dans l'hypothèse où un trinquage cordial existe, écrivez un algorithme qui indique les étapes à suivre afin d'effectuer le trinquage cordial, et indiquez sa complexité temporelle.

Algorithme 4 : Liste des actions pour effectuer un trinquage cordial

---

Entrées : Les sommets du graphe, S, c'est-à-dire l'ensemble des personnes impliquées
Les arêtes du graphe, A, c'est-à-dire l'ensemble des relations entre les individus
Résultat : L'ordre des trinquages dans un trinquage cordial

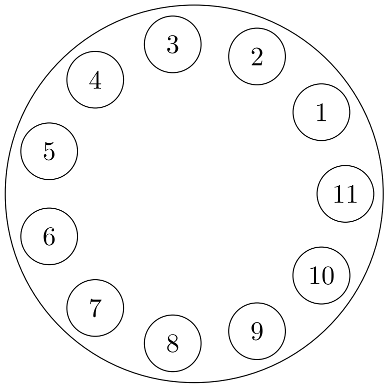
$N \longleftarrow |S|$ `// Le nombre de personnes`
`Adj` est un tableau de N listes, initialement vides ;
pour chaque *paire* (u, v) *dans* A **faire**
    Ajouter u à `Adj`$[v]$;
    Ajouter v à `Adj`$[u]$;
fin
`// Complétez le code`
`actions` est une liste de paires, initialement vide ;

retourner `actions` ;

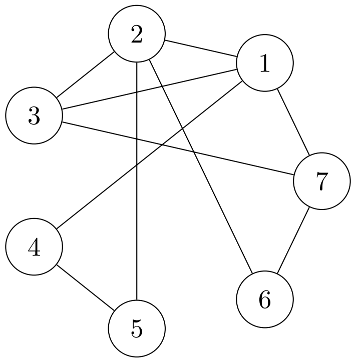
Question 8 (1 point)

Recherchez une manière permettant de trouver une répartition des individus sur la table ronde de l'Hersir.[9] respectant les RRLCDPDTRLRC. Cette table a pour particularité d'être ronde, et d'avoir un nombre impair de places. Chaque personne a donc 2 voisins, mais n'a personne en face. On doit donc simplement s'assurer que tout le monde possède une relation cordiale avec les personnes situées à leurs côtés.

Voici un exemple de table de 11 places :



Voici un graphe décrivant les relations cordiales entre 7 personnes.



Décrivez un arrangement de ces personnes sur une table de 7 places respectant les RRLCDPDTRLRC.

Question 9 (3 points)

Décrivez un algorithme résolvant cet exercice dans les grandes lignes, et indiquez sa complexité temporelle. En supposant que $P \neq NP$, pensez-vous qu'il est possible de résoudre ce problème en temps polynomial ?

Question 10 (2 points)

Un **graphe complet** est un graphe dont tous les sommets sont reliés à tous les autres.

Complétez l'algorithme suivant pour indiquer si un graphe est complet, et indiquez sa complexité temporelle :

Algorithme 5 : Vérification de la complétude d'un graphe

Entrées : Les sommets du graphe, S, c'est-à-dire l'ensemble des personnes impliquées
Les arêtes du graphe, A, c'est-à-dire l'ensemble des relations entre les individus
Résultat : `Vrai` si le graphe est complet, `Faux` sinon

$N \longleftarrow |S|$ `// Le nombre de personnes`
`tableau degré` $\longleftarrow$ Appel à `Algorithme1` ;
`// Complétez le code`

retourner *?* ;

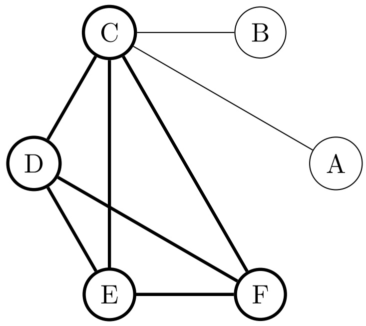
---

9. Pour ceux n'ayant pas révisé, un hersir est un commandant local viking.

Question 11 (2 points)

Définissons qu'un groupe d'amis est un ensemble de personnes (potentiellement vide) dont chaque individu du groupe est ami avec tous les autres individus [10].

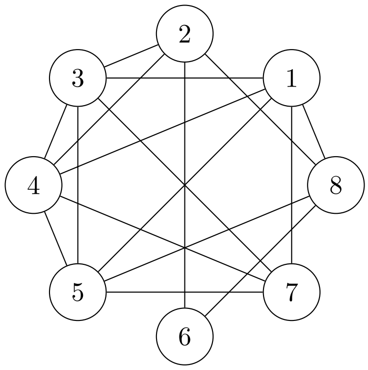
Dans le graphe suivant, le plus grand groupe d'amis est $\{C, D, E, F\}$:



Indiquez le nombre de groupes d'amis distincts dans ce graphe.

Question 12 (1 point)

Trouvez le plus grand groupe d'amis du graphe suivant :



Question 13 (3 points)

Décrivez un algorithme afin de trouver le plus grand groupe d'amis d'un graphe. Indiquez la complexité de votre solution. En supposant que $P \neq NP$, pensez-vous qu'il est possible de résoudre ce problème en temps polynomial ?

Question 14 (4 points)

Proposez des algorithmes afin de trouver un groupe d'amis de cardinalité minimale [11]. Indiquez la complexité temporelle et de mémoire de votre solution. Pensez-vous qu'il est possible de résoudre ce problème en temps polynomial ?

Algorithme 6 : Algorithme de recherche de clique minimale

Entrées : Les sommets du graphe, S, c'est-à-dire l'ensemble des personnes impliquées
Les arêtes du graphe, A, c'est-à-dire l'ensemble des relations entre les individus
Résultat : Un groupe d'amis de cardinalité minimale

```
// Complétez le code
```

retourner *?* ;

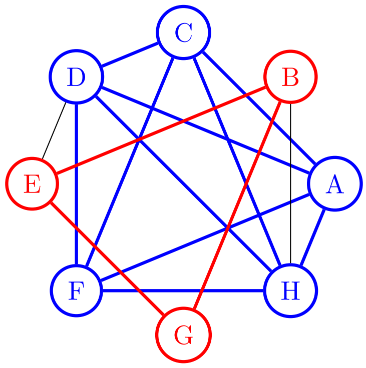
---

10. On appelle ça un « sous-graphe complet » ou une « clique ».
11. C'est-à-dire avec le moins de personnes

Question 15 (1 point)

Røger Perdu (un vieux savant malicieux du village) vient voir Jøsëf pour poser un problème : « Imaginons qu'on ait deux tables de taille indéterminée. Cependant, maintenant, sur chaque table, **tout le monde** doit s'entendre. Répartissez, si possible, tout le village sur ces deux tables. » [12]

Voici un exemple de graphe d'amitiés :



On peut alors placer :

- sur la première table : A, C, D, F et H ;
- sur la seconde table : B, E et G.

Proposez une répartition des sommets du graphe décrit dans la question 12.

Question 16 (4 points)

Rédigez un algorithme qui répartit, si possible, tous les habitants d'un village sur les deux tables, et décrivez sa complexité temporelle. En supposant que $P \neq NP$, pensez-vous qu'il est possible de résoudre ce problème en temps polynomial ?

Algorithme 7 : Partition du village sur deux tables de taille indéterminée

Entrées : Les sommets du graphe, S, c'est-à-dire l'ensemble des personnes impliquées
Les arêtes du graphe, A, c'est-à-dire l'ensemble des relations entre les individus

Résultat : Une partition des habitants en deux tables, où toutes les personnes d'une même table entretiennent une relation cordiale. Une erreur si cela est impossible.

```
// Construction de la liste d'adjacence
Adj est un tableau de N listes, initialement vides;
pour chaque paire (u, v) dans A faire
    Ajouter u à Adj_v ;
    Ajouter v à Adj_u ;
fin
Pour toute personne i dans S, table[i] ⟵ PAS_ASSIGNÉ;
// Complétez le code




retourner table;
```

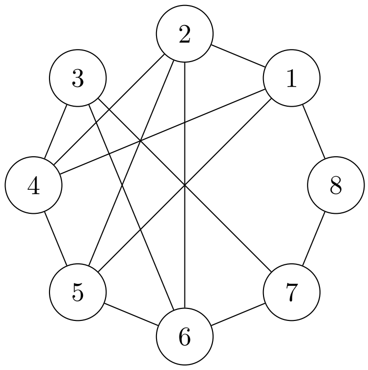
12. La légende dit que Røger est le créateur de Proidentifiant.

Question 17 (1 point)

Le problème posé par Røger trotte [13] toujours dans la tête de Jøsëf :

« Que se passe-t-il si on n'est pas forcément capable de répartir tout le monde sur deux tables ? »

S'il est impossible de répartir le village sur deux tables, proposez une répartition pour le village décrit par les relations suivantes, afin de déterminer le nombre minimum de tables nécessaires pour assigner une place à tous les habitants. Les tables peuvent toujours avoir la taille que l'on souhaite. Autour de chaque table, tout le monde doit avoir une entente cordiale avec tout le monde.



Question 18 (2 points)

S'il est impossible de répartir le village sur deux tables, proposez une technique pour déterminer le nombre minimum de tables nécessaires pour assigner une place à tous les habitants. Les tables peuvent toujours avoir la taille que l'on souhaite. Autour de chaque table, tout le monde doit avoir une entente cordiale avec tout le monde.

Décrivez votre technique dans les grandes lignes, analysez sa complexité temporelle. En supposant que $P \neq NP$, pensez-vous qu'il est possible de résoudre ce problème en temps polynomial ?

13. Pas littéralement bien évidemment, même si les deux options risquent de lui faire mal à la tête

4 Pot-au-flot

Définitions

— Un **graphe orienté** est un regroupement de **nœuds** qui peuvent être reliés entre eux par des **arcs**. Remarquez que le **nœud** et l'**arc** sont l'équivalent orienté du **sommet** et de l'**arête**.

— Un **arc** possède une direction, on peut aller du **nœud de départ** au **nœud d'arrivée** de l'**arc**.

— Un **réseau de flot** est un graphe orienté où les arcs sont pondérés avec une capacité. Cette capacité indique le débit maximal que peut accepter un arc.

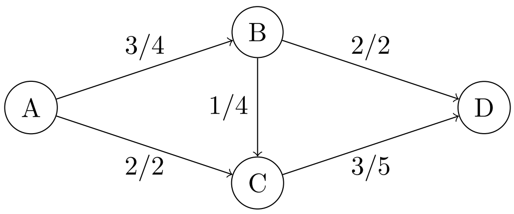
— Un arc d'un réseau de float est dit **saturé** lorsque son débit égale sa capacité. On ne peut donc plus ajouter de débit supplémentaire à travers cet arc.

— Le **flot** est le débit total allant de la source jusqu'au puits.

— La **source** est le nœud d'où proviennent les flots et le **puits** celui où finissent les flots. La source et le puits sont deux nœuds distincts.

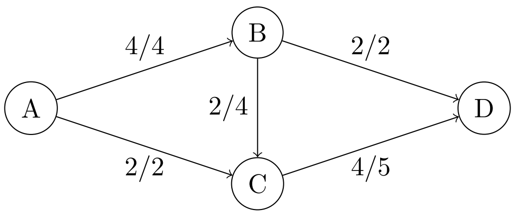
— Dans un réseau de flot, la somme des débits entrants à un nœud doit être égale à la somme des débits sortants.

On représente le débit et la capacité d'un arc de cette manière : *débit / capacité* :



Un **flot maximum** est un flot auquel on ne peut pas trouver un autre flot valide avec un débit strictement supérieur.

Voici le flot maximum (dont le débit total est $2 + 4 = 6$) du réseau ci-dessus :



4.1 Recettes

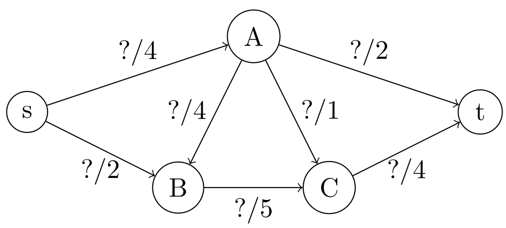
Maintenant que Jøsëf a rassuré Ødric et Jüliette, il remarque quelque chose : une fuite dans le réseau d'eau courante du village !

Le repas étant constitué de fondue auvergnate, tartiflette, raclette et fromages, il va être nécessaire d'avoir un grand débit d'eau pour hydrater les convives. Or, cette fuite nécessite de condamner le tuyau principal d'eau.

Pour que le débit reste suffisant, il va falloir faire passer l'eau par des chemins annexes ayant des tuyaux vétustes, limitant ainsi le débit maximal.

Question 19 (1 point)

Voici le réseau de secours permettant de fournir de l'eau potable à la fontaine de la place :

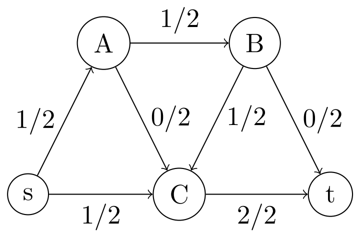


La source est s et le puits t.

Trouvez un flot maximum dans ce réseau.

Question 20 (1 point)

Trouvez un chemin dans le réseau de flot suivant, tel qu'aucun arc du chemin ne soit saturé.



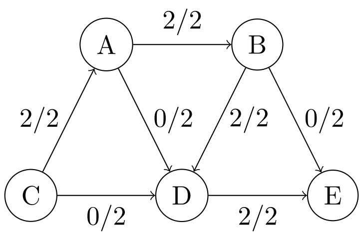
Question 21 (2 points)

Nouvelle règle ! Trouvez un chemin dans le réseau de flot suivant, en passant soit :

— par des arcs du flot qui ne sont pas saturés,

— par des arcs du flot dont le débit n'est pas nul, en sens inverse

Puis, proposez une manière d'augmenter le flot en modifiant uniquement les débits des arcs sur le chemin trouvé. [14]



Question 22 (3 points)

Proposez un algorithme permettant de trouver un chemin augmentant dans un réseau de flot, et indiquez sa complexité temporelle.

Algorithme 8 : Recherche de chemin augmentant dans un réseau de flot

---

Entrées : Les sommets du graphe, S, dont deux sommets s et t, respectivement la source et le puits
Les arcs du graphe, A, ainsi qu'une application $c(x)$ indiquant la capacité de l'arc x, et une application $d(x)$ indiquant le débit traversant l'arc x
Résultat : Un chemin augmentant dans le flot

`Adj` est un tableau de N listes, initialement vides ;
`chemin` est une liste vide ;
`// Code à compléter`

retourner `chemin` ;

Question 23 (3 points)

Proposez un algorithme trouvant le flot maximum d'un réseau. Indiquez sa complexité temporelle.

Question 24 (2 points)

Est-il possible d'avoir plusieurs flots maximum dans un réseau de flot ? Justifier.

14. On appelle un tel chemin un « chemin augmentant »

Question 25 (3 points)

Indiquez une stratégie réutilisant votre algorithme de la question 23 afin de supporter un réseau avec plusieurs puits ou [15] plusieurs sources.

5 Bonus

Question bonus 26 (1 point)

Écrivez l'intégralité de vos réponses en bon Français.
N'utilisez aucun anglicisme ou mot anglais dans votre sujet.

Question bonus 27 (1 point)

Proposez une blague en expression régulière.

Question bonus 28 (1 point)

Dessinez les plans du futur siège de Proidentifiant lorsque l'association fera partie du CAC40.

Question bonus 29 (1 point)

Répondez à une seule de ces questions, au choix :

- Quelle est la taille minimale pour participer au concours ?
- Dessinez le graphe de dépendance des définitions incluses dans le sujet.

Question bonus 30 (1 point)

Soit Θ un ensemble.
Voici Grégoire : $\bar{\Theta} \subseteq \bar{\Theta}$
Proposez un ami à Grégoire qui est faux.

15. Ou inclusif