TD 1 : Modélisations (correction)

Exercice 1

1. On compte :
   - les boites de petites allumettes x_{Ap},
   - les boites de grandes allumettes x_{Ag},

   ▷ **Remarque :** On pourrait parler de lot de 100 000 allumettes, cf. question 6.
2. On veut maximiser le profit, c'est-à-dire $300 \times x_{Ag}/100\,000 + 200 \times x_{Ap}/100\,000$.
3. $3 \times x_{Ag}/100\,000 + x_{Ap}/100\,000 \leq 18$.
4. $x_{Ag} + x_{Ap} \leq 900\,000$.
5. $x_{Ap} \leq 600\,000$ et $x_{Ag} \leq 700\,000$.
6. Les variables sont entières positives. Elles doivent même être multiples de 100 000. On pourrait tout réécrire en posant $x'_{Ap} = x_{Ap}/100\,000$ et $x'_{Ag} = x_{Ag}/100\,000$.

Exercice 2

▷ **Méthode :**

1. Quelles sont les données ?
2. Que veut-on maximiser ?
3. Comment calculer simplement ce qu'on veut maximiser ?

On a 7 données mises en jeu : la quantité de whisky utilisée,la quantité de vodka utilisée, la quantité de gin utilisée, le nombre de Manhattan produit, le nombre de Harlem produit et le nombre de Spécial produit, l'argent gagné.

Pour connaitre le variables du LP, on regarde ce qu'il faut maximiser. Il s'agit de l'argent gagné. Comment calculer facilement l'argent gagné et en fonction de quelles variables ? L'argent gagné est le nombre de Manhattan produit fois 5 \$, plus le nombre de Harlem produit fois 4 \$, plus le nombre de Spécial produit fois 6 \$.

Donc les variables du LP sont x_m le nombre de Manhattan produit, x_h le nombre de Harlem produit et x_s le nombre de Spécial produit.

On maximise : $5x_m + 4x_h + 6x_s$, sous les contraintes :

$$10x_m + 10x_s \leq 50$$
$$5x_m + 5x_h \leq 50$$
$$10x_h + 15x_s \leq 100$$

Exercice 3 **(A) Échanges linguistiques**

1. Un sommet représente un étudiant. Deux étudiants ont reliés par une arête s'ils parlent une langue en commun. On cherche une clique maximum. On peut aussi faire l'inverse : une arête s'il ne parlent pas de langue en commun, et on cherche un ensemble indépendant maximum.
2. Une variable x_i par étudiant. Une contrainte $0 \leq x_i \leq 1$ par étudiant (ainsi, $x_i \in \{0, 1\}$: 0 s'il n'est pas invité, 1 s'il l'est). Pour chaque pair d'étudiants x_i, x_j qui ne parle aucune langue en commun, $x_i + x_j \leq 1$ (cela signifie qu'un seul des étudiants pourra être invité). On maximise $\sum_i x_i$, c'est-à-dire le nombre d'invité.
3. En modélisant par des ensembles indépendants, on cherche une coloration utilisant le moins de couleur possible.

(B) Emplois du temps

1. Sommets : c_i pour les cours. Arêtes : une arête entre c_i et c_j s'il y a un étudiant inscrit à la fois aux cours c_i et c_j. Problème : coloration.
2. Sommets x_i pour les cours. Arêtes : une arête entre x_i et x_j si les cours x_i et x_j se chevauchent. Problème : coloration.

(C) Journée des sports

Sommets : x_i pour chaque sport. Arêtes : une arête entre x_i et x_j si les sports x_i et x_j sont en conflit. Problème : ensemble indépendant maximum.

(D) Problème de transport

▷ **Remarque :** Il manquait un problème de graphe dans l'énoncé : « Trouver un couplage de taille maximale dans un graphe biparti pondéré. En cas d'égalité, on cherchera à avoir le couplage de poids minimal, c'est-à-dire pour lequel la somme des poids des arêtes choisies est le plus petit possible. »

On commence par découper chaque entrepôt de capacité r_i en r_i entrepôts de capacité 1. Comme ça, chaque entrepôt a exactement un téléphone. On crée un graphe biparti pondéré : la partie gauche du graphe contient un sommet x_i par téléphone (c'est-à-dire par entrepôt, puisque maintenant, chaque entrepôt contient exactement un téléphone). La partie droite contient un sommet y_j par client. Il y a une arête entre chaque paire de sommets (x_i, y_j), avec un poids $c_{i,j}$.

Exercice 4

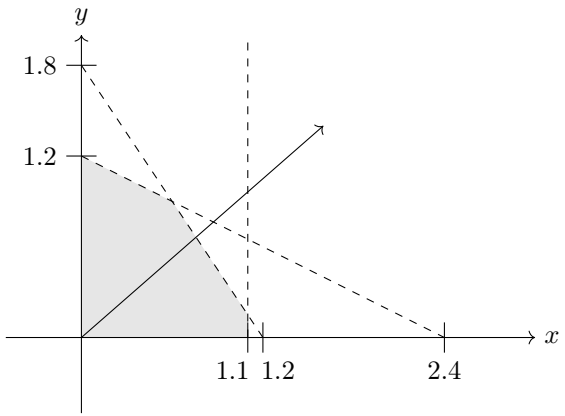
1. $x_{i,j}$ pour $1 \leq i \leq n, 1 \leq j \leq m$ qui indique si le client C_j a reçu son smartphone depuis l'entrepôt X_i.
2. pour tout j : $\sum_i x_{i,j} = 1$.
3. pour tout i : $\sum_j x_{i,j} \leq r_i$.
4. $\min \sum_{i,j} x_{i,j} r_{i,j}$.

TD 2 : Programmation linéaire (correction)

Exercice 1 Je note x le nombre de paquets de Frisky Pup, y le nombre de paquets de Husky Hound. On veut optimiser $(7-1.4-1-3)x+(6-0.6-2-2)y = 1.6x+1.4y$. On a les contraintes :

$$\begin{aligned} x + 2y &\leq 240\ 000 \\ 1.5x + y &\leq 180\ 000 \\ x &\leq 110\ 000 \end{aligned}$$

On obtient la figure (on a divisé les unités par 100 000) :



La solution se trouve dans la zone grise. Par théorème, c'est un sommet du polytope. On essaie les 5 sommets et on prend le meilleur, ou on voit lequel est le plus dans la direction de la flèche :

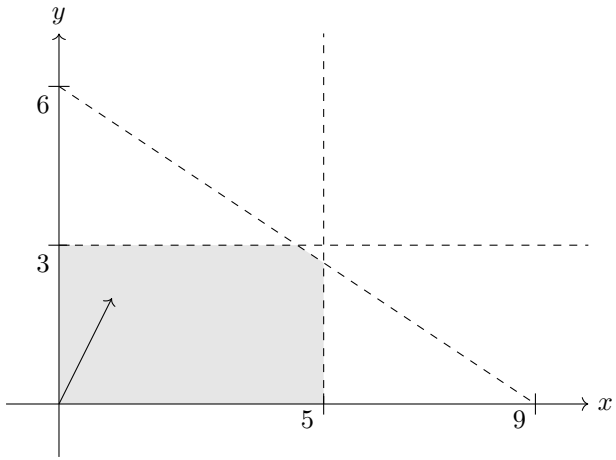
- $x = 0$ et $y = 0$ donnent $1.6x + 1.4y = 0$ €,
- $x = 0$ et $y = 120\ 000$ donnent $1.6x + 1.4y = 168\ 000$ €,
- $x = 60\ 000$ et $y = 90\ 000$ donnent $1.6x + 1.4y = 222\ 000$ €,
- $x = 110\ 000$ et $y = 15\ 000$ donnent $1.6x + 1.4y = 197\ 000$ €.
- $x = 110\ 000$ et $y = 0$ donnent $1.6x + 1.4y = 176\ 000$ €.

Exercice 2

1. Je note x la quantité de soupe produite en hL, y la quantité d'huile produite en hL. On veut optimiser $x + 2y$, où les coefficients expriment des k€. On a les contraintes :

$$\begin{aligned} 2x + 3y &\leq 18 \\ y &\leq 3 \\ x &\leq 5 \end{aligned}$$

2. On obtient la figure :



La solution se trouve dans la zone grise. Par théorème, c'est un sommet du polytope. On pourrait essayer les 5 sommets, mais vu la direction de la flèche, 2 semblent être plus prometteurs :

- $x = 5$ et $y = 8/3 \approx 2.3333$ donnent $x + 2y = 5 + 4.6666 = 9.6666$ k€,
- $x = 4.5$ et $y = 3$ donnent $x + 2y = 4.5 + 6 = 10.5$ k€.

Exercice 3

▷ **Comprendre le dual sur un exemple :** Considérons le PL suivant.

$$\begin{aligned} \max\ & 3x_1 + 2x_2 & \\ & x_1 \leq 2000 & \text{(Contrainte 1)} \\ & x_2 \leq 3000 & \text{(Contrainte 2)} \\ & 2x_1 + 3x_2 \leq 7000 & \text{(Contrainte 3)} \end{aligned}$$

Je cherche une borne triviale sur le maximum de ce PL. Si je fais $3 \times$ (Contrainte 1) $+$ $2 \times$ (Contrainte 2), j'obtiens $2x_1 + 3x_2 \leq 12000$. Si je fais $0.5 \times$ (Contrainte 3) $+$ $2 \times$ (Contrainte 1) $+ 0.5 \times$ (Contrainte 2), j'obtiens $3x_1 + 2x_2 \leq 9000$. Je cherche

des coefficients y_1, y_2, y_3 tels que $y_1 \times$ (Contrainte 1) + $y_2 \times$ (Contrainte 2 + $y_3 \times$ (Contrainte 3) donne la meilleure borne possible. Pour que les coefficients donnent une borne, il faut que j'ai assez de x_1 (au moins $3x_1$), donc que $y_1 + 2y_3 \geq 3$, et assez de x_2 : chaque variable du LP donne une contrainte. De plus, chaque $y_i \geq 0$, car sinon, cela change le sens de l'inégalité et ça ne donne pas de borne supérieure.

Pour info : $x_1 = 2000, x_2 = 1000, y_1 = 5/3, y_2 = 0, y_3 = 2/3$.

▷ **Méthode :**

— Je mets le LP sous forme canonique $Ax \leq b$: les contraintes $\geq$ deviennent $\leq$ en multipliant par -1 et les contraintes $=$ sont scindées en $\geq$ et $\leq$.

— $Ax \leq b$ soumis à $\max c^T.x$ devient $Ay \geq c$ soumis à $\min b^T.y$.

Exercice 4

1. On pose une variable x_v pour chaque sommet $v \in V$:

$$\begin{aligned} & \max \sum_{v \in V} x_v \\ \forall \{u,v\} \in E : \quad & x_u + x_v \leq 1 \\ \forall v \in V : \quad & x_v \leq 1 \\ \forall v \in V : \quad & x_v \geq 0 \end{aligned}$$

2. $1 - 0 - 0$.
   $0.5\times$(Contrainte 1)$+0.5\times$(Contrainte 2$+0.5\times$(Contrainte 3) donne $x_1+x_2+x_3 \leq 3/2$. Comme c'est entier, $x_1 + x_2 + x_3 \leq 1$.
3. $1/2 - 1/2 - 1/2$. Comme la question précédente. C'est différent de la solution entière.
4. On peut toujours mettre $x_v = 1/2$ à tous les sommets (et même $x_v = 1$ si le sommet est isolé.) Donc $P^*_f \geq n/2$. Dans une clique, $P^* = 1$. D'où la famille de cliques.

Exercice 5

▷ **Méthode :**

— Pour chaque contrainte, je remplace les x_i par leur valeur et je regarde comment se comporte l'inégalité : si elle n'est pas remplie, la solution n'est pas admissible, donc pas optimale ; si c'est une égalité, on dit que la contrainte est saturée, rien à en déduire ; si l'inégalité est stricte, on dit que la contrainte est non-saturée, on en déduit que le y_j correspondant vaut 0.

— Pour chaque variable $x_i > 0$, la contrainte correspondante dans le dual doit être une égalité.

— On résout le système obtenu. S'il n'y a pas de solution, ce n'est pas optimal. Sinon, elle est optimale si et seulement si elle est admissible.

(a) Le dual est :

$$\begin{aligned} & \min 4y_1 + 12y_2 + y_3 \\ & y_1 + 4y_2 - y_3 \geq 1 \\ & 2y_1 + 2y_2 + y_3 \geq 1 \\ & y_1, y_2 \geq 0 \end{aligned}$$

1. Dans le primal, la contrainte 2 n'est pas saturée, donc $y_2 = 0$. On a $x_1 > 0$ et $x_2 > 0$, donc on résout :

$$\begin{aligned} y_1 + 4 \times 0 - y_3 &= 1 \\ 2y_1 + 2 \times 0 + y_3 &= 1 \end{aligned}$$

   On obtient $y_1 = 2/3$ et $y_3 = -1/3$. On a bien les objectifs qui satisfont $x_1 + x_2 = 7/3 = 4y_1 + 12y_2 + y_3$. C'est optimal, mais en fait y_3 négatif interdit la solution. ($y_3 \geq 0$ devrait apparaître dans le dual).
2. Solution non admissible.
3. C'est optimal.

(b) Le dual est :

$$\begin{aligned} & \min 5y_3 \\ & -y_1 + y_2 + y_3 \geq 1 \\ & 3y_1 - 4y_2 + 2y_3 \geq 7 \\ & -2y_1 + 2y_2 + 3y_3 \geq 3 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}$$

Dans le primal, la contrainte 2 n'est pas saturée, donc $y_2 = 0$. On a $x_1 > 0$ et $x_2 > 0$, donc on résout :

$$\begin{aligned} -y_1 + y_3 &= 1 \\ 3y_1 + 2y_3 &= 7 \\ -2y_1 + 3y_3 &\geq 3 \end{aligned}$$

On obtient $y_1 = 1$ et $y_3 = 2$. On a bien $-2y_1 + 3y_3 = 4 \geq 3$. C'est optimal.

Exercice 6

1. Je me le PL (P) sous forme $\max c^T.x$ soumis à $Ax \leq b$:

$$\begin{aligned} & \max -30x_1 + x_2 - 3x_3 - 5x_4 \\ & -4x_1 + x_2 - x_3 + 0x_4 \leq -5 \\ & -5x_1 + x_2 + x_3 - x_4 \leq -3 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Le dual est :

$$\begin{aligned} \min\ & -5y_1 - 3y_2 \\ & -4y_1 - 5y_2 \geq -30 \\ & y_1 + y_2 \geq 1 \\ & -y_1 + y_2 \geq -3 \\ & -y_2 \geq -5 \\ & y_1, y_2 \geq 0 \end{aligned}$$

c'est-à-dire :

$$\begin{aligned} \max\ & 5y_1 + 3y_2 \\ & 4y_1 + 5y_2 \leq 30 \\ & y_1 + y_2 \geq 1 \\ & y_1 - y_2 \leq 3 \\ & y_2 \leq 5 \\ & y_1, y_2 \geq 0 \end{aligned}$$

2. Le dual (D), car il n'y a que 2 variables, on peut donc le résoudre graphiquement.
3. ▷ **Méthode (adaptée à ce cas-là) :** Je trace le tableaux suivant :

	Variable = 0	Contrainte =
x_1		
x_2		
x_3		
x_4		
y_1		
y_2		

Le Théorème des écarts complémentaires me dit que si (x_1, x_2, x_3, x_4) est une solution admissible de (P) et (y_1, y_2) est une solution admissible de (D), alors chaque ligne de mon tableau doit avoir une case cochée, c'est-à-dire que soit la variable vaut 0, soit la contrainte associée dans l'autre LP est une égalité. J'essaie de remplir ce tableau afin que mes solutions soient optimales, sans oublier de vérifier qu'elles sont bien admissibles.

Les contraintes donnent :

$$4x_1 - x_2 + x_3 = 0 + 0 + 5 = 5 \geq 5$$
$$5x_1 - x_2 - x_3 + x_4 = 0 + 0 - 5 + 8 = 3 \geq 3$$

Le tableau donne :

	Variable = 0	Contrainte =
x_1	✓	
x_2	✓	
x_3	×	
x_4	×	
y_1		✓
y_2		✓

Les contraintes sont vérifiées et saturées. Or, comme $x_3 > 0$ et $x_4 > 0$, on en déduit que les 2 inégalités issues de x_3 et x_4 doivent être des égalités (pour pouvoir mettre des ✓ dans le tableau) :

$$\begin{aligned} y_1 - y_2 &= 3 \\ y_2 &= 5 \end{aligned}$$

Cela donne $y_1 = 8$. Le tableau donne :

	Variable = 0	Contrainte =
x_1	✓	
x_2	✓	
x_3	×	✓
x_4	×	✓
y_1		✓
y_2		✓

Cette solution n'est pas admissible, elle n'est donc pas optimale.

Exercice 7

1. On pose une variable x_v pour chaque sommet $v \in V$. Le PLNE est :

$$\begin{aligned} & \min \sum_{v \in V} x_v \\ \forall e \in E : \quad & \sum_{v \in e} x_v \geq 1 \\ \forall v \in V : \quad & x_v \leq 1 \\ \forall v \in V : \quad & x_v \geq 0 \end{aligned}$$

2. Avant de construire le dual, mettons le PL sous forme $Ax \leq b \mid \max c^T.x$:

$$\begin{array}{lr} & \max \sum_{v\in V} -x_v \\ \forall e \in E : & \sum_{v\in e} -x_v \leq -1 \\ \forall v \in V : & x_v \leq 1 \\ \forall v \in V : & x_v \geq 0 \end{array}$$

Le dual est (on pose y_e pour tout $e \in E$ et z_v pour tout $v \in V$) :

$$\begin{array}{lr} & \min \sum_{e\in E} -y_e + \sum_{v\in V} z_v \\ \forall v \in V : & \sum_{v\in e\in E} -y_e + z_v \geq -1 \\ \forall e \in E : & y_e \geq 0 \\ \forall v \in V : & z_v \geq 0 \end{array}$$

On met sous forme plus naturelle :

$$\begin{array}{lr} & \max \sum_{e\in E} y_e - \sum_{v\in V} z_v \\ \forall v \in V : & \sum_{v\in e\in E} y_e - z_v \leq 1 \\ \forall e \in E : & y_e \geq 0 \\ \forall v \in V : & z_v \geq 0 \end{array}$$

On se rend compte que l'on ne peut pas conclure à cause des z_v. Recommençons. On peut en fait supprimer les contraintes $x_v \leq 1$ du primal. En effet, si j'autorise des x_v valant 2 ou plus, je n'autorise que des solutions avec une valeur objectif plus grande. On a donc le primal :

$$\begin{array}{lr} & \max \sum_{v\in V} -x_v \\ \forall e \in E : & \sum_{v\in e} -x_v \leq -1 \\ \forall v \in V : & x_v \geq 0 \end{array}$$

D'où le dual :

$$\begin{array}{lr} & \max \sum_{e\in E} y_e \\ \forall v \in V : & \sum_{v\in e\in E} y_e \leq 1 \\ \forall e \in E : & y_e \geq 0 \end{array}$$

Dans le cas où $\emptyset \notin E$, c'est bon.

3. Dualité forte : ils ont même taille.

Exercice 8

1. $V = [n]$ et $i \to j$ dès que $i < j$.
2. On montre que ce tournoi à n sommets est celui décrit à la question 1. On le fait par récurrence sur n : si c'est vrai pour n, on prend un tournoi à $n+1$ sommets, on retire un sommet u, on applique H_r, ce qui nous dit que l'on peut renommer ces sommets en $1, \ldots, n$ et que $i \to j$ ssi $i < j$.

   On regarde comment u est relié aux sommets $1, \ldots, n$. Je prétends qu'il existe k tel que $i \to u$ pour $i < k$ et $u \to i$ sinon. En effet, si ce n'est pas le cas, on a $u \to i \to j \to u$ $(i < j)$, ce qui constitue un triangle orienté.

   On renomme u en k et tous les sommets $i \geq k$ en $i + 1$.
3. On parcourt tous les triplets $i < j < k$.
4. On pose une variable x_s pour chaque triangle orienté $s \in \mathcal{S}$. Le PLNE (P) est :

$$\begin{array}{lr} & \max \sum_{s\in\mathcal{S}} x_s \\ \forall v \in V : & \sum_{v\in s\in\mathcal{S}} x_s \leq 1 \\ \Big(\forall s \in \mathcal{S} : & x_s \leq 1\Big) \\ \forall s \in \mathcal{S} : & x_s \geq 0 \end{array}$$

▷ **Remarque :** Quand on demande un LP, on peut mettre autant de contrainte que l'on veut. Mais si ensuite on demande le dual, il vaut mieux en mettre le moins possible, car chaque contrainte donne une variable dans le dual.

5. On pose une variable y_v pour chaque sommet $v \in V$. Le PLNE (D) est :

$$\begin{aligned} & \min \sum_{v \in V} y_v \\ \forall s \in \mathcal{S} : \quad & \sum_{v \in s} y_v \geq 1 \\ \Big(\forall v \in V : \quad & y_v \leq 1\Big) \\ \forall v \in V : \quad & y_v \geq 0 \end{aligned}$$

6. On note P_f^* la valeur optimale de la relaxation fractionnaire du PL (P). Idem pour les 3 autres valeurs optimales. Par dualité forte, $P_f^* = D_f^*$. Puisque (P) est un problème de maximisation, la relaxation fractionnaire ne peut que donner une valeur plus grande. D'où $P^* \leq P_f^*$. Puisque (D) est un problème de minimisation, la relaxation fractionnaire ne peut que donner une valeur plus petite. D'où $D^* \geq D_f^*$. Finalement, $P^* \leq P_f^* = D_f^* \leq D^*$.
7. Notons X cet ensemble. Deux choses à montrer : X est un feedback vertex set, $|X| \leq 3D^*$.
   — Comme l'ensemble des triangles orientés choisis S est maximal pour l'inclusion, tout triangle orienté partage un sommet avec un triangle orienté de S. Soit T un triangle orienté, il partage un sommet avec un triangle de S, c'est-à-dire un sommet de X. Donc tout triangle orienté a un sommet dans X. Donc X est un feedback vertex set.
   — Soit S l'ensemble des triangles orientés choisis. On a $|S| \leq P^*$ puisque P^* est la quantité maximale. Donc $|S| \leq D^*$ d'après la question précédente. Enfin, $|X| = 3|S| \leq 3D^*$.
8. La question 3 permet d'avoir la liste des triangles orientés, puis la question 7 permet de construire X souhaité.

Exercice 9

1. On pose une variable x_c pour la quantité de colliers, une variable x_b pour la quantité de bracelets et une variable x_o pour la quantité de boucles d'oreilles :

$$\begin{aligned} \max\, & 8x_c + 5x_b + 3x_o \\ & 8x_c + 4x_b + 2x_o \leq 15\ 000 \\ & 0x_c + 4x_b + 6x_o \leq 30\ 000 \\ & x_c, x_b, x_0 \geq 0 \end{aligned}$$

2. Le dual est :

$$\begin{aligned} \min\, & 15\ 000y_1 + 30\ 000y_2 \\ & 8y_1 + 0y_2 \geq 8 \\ & 4y_1 + 4y_2 \geq 5 \\ & 2y_1 + 6y_2 \geq 3 \\ & y_1, y_2 \geq 0 \end{aligned}$$

3. Avant d'appliquer bêtement 2 écarts complémentaires, regardons si les solutions sont admissibles et calculons leur fonction objectif. Les deux solutions sont admissibles La solution $(625, 0, 5000)$ donne un objectif de 20 000 \$. La solution $(0, 1875, 3750)$ donne un objectif de 20 625 \$. Donc la solution $(625, 0, 5000)$ n'est pas optimale. On applique la méthode des écarts complémentaires pour la deuxième solution :
   Après substitution :

$$\begin{aligned} 8x_c + 4x_b + 2x_o &= 15\ 000 \leq 15\ 000 \\ 0x_c + 4x_b + 6x_o &= 30\ 000 \leq 30\ 000 \end{aligned}$$

   Les deux inégalités sont vérifiées et saturées. Le tableau donne :

	Variable = 0	Contrainte =
x_c	✓	
x_b	×	
x_o	×	
y_1		✓
y_2		✓

   Il faut :

$$\begin{aligned} 8y_1 + 0y_2 &\geq 8 \\ 4y_1 + 4y_2 &= 5 \\ 2y_1 + 6y_2 &= 3 \\ y_1, y_2 &\geq 0 \end{aligned}$$

   Cela donne $y_1 = 3/4$ et $y_2 = 1/2$. La solution est admissible, elle est optimale.

TD 3 : Algorithmes gloutons et d'approx. (correction)

Exercice 1

1. Supposons par l'absurde que I n'est pas indépendant, c'est-à-dire qu'il existe une arête uv avec $u, v \in I$. Par symétrie, on peut supposer que v est considéré après u dans l'algorithme. En particulier, quand v est considéré, il est adjacent au sommet u déjà dans I et on n'aurait donc pas dû le prendre. Contradiction. Ainsi, I est indépendant.

   Supposons maintenant par l'absurde qu'il existe un ensemble indépendant J tel que $I \subsetneq J$. Soit $u \in J \setminus I$. Au moment où l'algorithme glouton a considéré u, il doit le mettre dans I, vu qu'il n'a aucun voisin dans $J \supset I$. Contradiction.

2. Les opérations d'ajout coûtent $O(1)$ chacune (si I est représenté par un tableau de bits), donc au total elle représentent $O(n)$. Pour la condition à tester, il faut, pour chaque sommet u, parcourir ses voisins v et tester si $v \in I$. Chaque test se fait en $O(1)$. Au total, on effectue $O(1)$ opérations pour chaque arête incidente à u. Chaque arête du graphe apparaît deux fois dans ce calcul (une pour chaque extrémité). La complexité finale est donc $O(n+m)$.

3. Non. Exemple : $b - a - c$ et on considère les sommets dans l'ordre alphabétique.

4. Considérons un graphe à n sommets en étoile : un sommet central et $n-1$ feuilles. L'algorithme glouton (s'il commence par le sommet central) renvoie un ensemble indépendant de taille 1, alors que l'ensemble maximum est de taille $n-1$ (en prenant toutes les feuilles). On ne peut pas espérer un pire écart (sinon, on devrait considérer un graphe avec un ensemble indépendant de taille n, c'est-à-dire sans arête, mais dans ce cas l'algorithme glouton trouve cet ensemble indépendant).

Exercice 2

1. Non : pour T_3, on a colorié d avec la couleur 3 alors que le graphe est biparti, donc coloriable avec 2 couleurs.

2. Initialisation : v_1 a pour couleur $1 = 0 + 1 \leq \Delta + 1$.

   Hérédité : C'est vrai pour les k premiers sommets visités. Soit v_{k+1}. On considère ses $\delta_- \leq \Delta$ voisins vu avant lui. Comme il y a $\Delta + 1$ couleurs possibles pour v_{k+1} et qu'il y a au plus $\delta_- \leq \Delta$ couleurs interdites par ses voisins, au moins une couleur $c \leq \Delta + 1$ est disponible. Comme l'algorithme prend la plus petite couleur disponible, elle vérifie $c \leq \Delta + 1$.

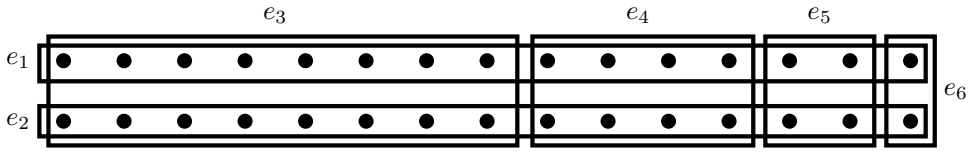
3. On remarque que T_4 est construit en prenant un sommet v_1 et en le reliant aux racines de T_1, T_2, T_3. On colorie d'abord T_1, T_2 et T_3 dans l'ordre proposé à la question 1 pour que leur racine soit respectivement colorié 1, 2 et 3. On colore v_1 en dernier et la couleur doit être 4. Un exemple d'ordre de ce type : $v_2, v_5, v_3, v_6, v_8, v_7, v_4, v_1$.

4. T_i vaut v_1 relié aux racines de $T_1, \ldots, T_{i-1}$. En notant σ_i l'ordre de visite de T_i, on a $\sigma_i = \sigma_1 \cdots \sigma_{i-1} \cdot v_1$. En suivant cet ordre, on force le premier fils de la racine à être coloré 1, le second avec 2, ..., le $(i-1)$-ème avec $i-1$. La racine doit ensuite être colorée i.

5. Soit $c : S \to \mathbb{N}^*$ une coloration optimale. Je considère un ordre qui place d'abord les sommets de couleur 1, puis ceux de couleur 2, etc. dans l'ordre jusqu'à la couleur maximale χ. Je montre par récurrence que chaque sommet v sera colorié par l'algorithme glouton par une couleur inférieure ou égale à $c(v)$: au moment de colorier les sommets u tels que $c(u) = i$, tous les autres sommets seront soit coloriés par des couleurs $< i$, soit non-colorié. Comme ces sommets u forment un ensemble indépendant, aucun sommet à colorier avec une couleur $\leq i$ ne sera voisin d'un autre sommet à colorier avec une couleur $\leq i$, donc l'algorithme va tous les colorier avec une couleur $\leq i$. En particulier, à la fin, tous les sommets ont reçu une couleur $\leq \chi$.

Exercice 3

1. e_1, e_2.
2. Soit S un set cover optimal. Chaque arête de S couvre au plus deux sommets, donc le nombre total de sommets couverts est inférieur à $2|S|$. Comme c'est un set cover, tous les sommets sont couverts, donc $n \leq 2|S|$, autrement dit $|S| \geq n/2$.
3. e_3, e_4, e_5.
4. On propose :



5. On note $\mathcal{H}_p$ l'hypergraphe obtenu en généralisant la construction précédente. Plus précisément, il contient deux hyperarêtes disjointes e_1, e_2 de taille $2^p - 1$, et p hyperarêtes $f_1, \ldots, f_p$, deux à deux disjointes, telles que f_i contient précisément 2^{p-i-1} sommets de e_1 et 2^{p-i-1} sommets de e_2. Par construction f_i a taille 2^{p-i}. Cet hypergraphe contient $2 \times (2^0 + 2^1 + \ldots 2^{p-1}) = 2(2^p - 1)$ sommets, possède un set cover $\{e_1, e_2\}$ de taille 2, et l'algorithme glouton renvoie $\{f_1, \ldots, f_p\}$. En

effet, la plus grosse hyperarête est f_1, et retirer ses sommets donne l'hypergraphe $\mathcal{H}_{p-1}$.

Si on avait une α-approximation, on aurait $|S_{approx}| \leq \alpha |S_{opti}|$ pour toute entrée, en particulier ce serait le cas pour $\mathcal{H}_p$, et donc on a $p \leq \alpha 2$, c'est-à-dire $\alpha \geq p/2$. Comme $n = 2(2^p - 1)$, on a $p = \log_2(\frac{n}{2} + 1)$. D'où $\alpha \geq \frac{1}{2}\log_2(\frac{n}{2}+1) \geq \frac{1}{2}\log_2 \frac{n}{2} = \frac{1}{2}(\log_2 n - 1)$.

6. Je pose une variable x_i pour chaque hyper-arête e_i, en imaginant $e_i = 1$ si et seulement si e_i est sélectionné. On a :

$$\begin{aligned} & \min \sum_{i=1}^{|E|} x_i \\ \forall u \in V : \quad & \sum_{i|u\in e_i} x_i \geq 1 \\ \Big(\forall i : \quad & x_i \leq 1\Big) \\ \forall i : \quad & x_i \geq 0 \end{aligned}$$

7. On a une variable y_j par contrainte, c'est-à-dire par sommet :

$$\begin{aligned} & \max \sum_{j=1}^{n} y_j \\ \forall i : \quad & \sum_{j|u_j\in e_i} y_j \leq 1 \\ \forall j : \quad & y_j \geq 0 \end{aligned}$$

Le problème consiste à trouver le plus grand ensemble de sommets tel que chaque sommet appartient à au plus une hyperarête. Dans un graphe, ce serait un ensemble indépendant.

Exercice 4

1. On fait facilement une coupe de taille 3 avec $A = \{\text{Robin}\}$, de taille 4 avec $A = \{\text{Robin, Usopp}\}$, de taille 5 avec $A = \{\text{Robin, Zoro}\}$, de taille 6 avec $A = \{\text{Robin, Sanji}\}$. Il n'y a pas de coupe de taille 7, et 6 est donc l'optimal. En effet, comme il y a 8 arêtes dans le graphes, si on avait une coupe de taille 7, alors il n'y aurait qu'une arête qui ne serait pas coupée ; dans ce cas , l'un des deux ensemble de sommet (A ou $\overline{A}$) serait un ensemble indépendant, ce qui n'arrive que pour des cas déjà étudié (un coin seul ou deux coins opposés).
2. On a une coupe de taille m, le nombre d'arêtes. En effet comme le graphe est biparti, en coupant entre les deux parties, on coupe toutes les arêtes.
3. Elle s'exécute au pire m fois (une fois par d'arêtes), car à chaque déplacement, la valeur de la coupe augmente de 1 et elle est majorée par m.
4. La boucle est exécutée m fois. On calcule la complexité d'un tour de boucle. On imagine que l'on a une structure d'ensemble dans laquelle l'ajout, la suppression et le test d'appartenance sont en $O(1)$. Pour savoir s'il y a un sommet déplaçable, on parcourt chaque sommet pour savoir s'il est déplaçable. Pour savoir si un sommet donné est déplaçable, il faut calculer la valeur d'une coupe, ce qui peut se faire en parcourant chaque arête en regardant si ses extrémités sont chacune dans une partie différente de la coupe (l'une dans A et l'autre dans $\overline{A}$). On a donc une complexité totale de $O(nm^2)$, avec n le nombre de sommets.

   On peut faire mieux : quand je recalcule la coupe, la majorité des arêtes n'ont pas changé. Seules les arêtes incidentes au sommet v que je déplace bougent. Donc le test pour un sommet v coûte $d(v)$ (son degré ; voir question suivante), et le coût pour savoir s'il existe un sommet déplaçable est $\sum \max(1, d(v)) = O(n + m)$. D'où une complexité en $O(m(n + m))$.

   On peut faire encore mieux, en calculant d'abord tous les degrés des sommets (un preprocessing qui prend $O(n+m)$). Ensuite pour rechercher sommet déplaçable, on a juste à comparer les degrés calculés pour chacun, ce qui prend $O(n)$ au total. Pour que cela fonctionne, il faut pouvoir mettre à jour les degrés après déplacement, ce qui prend $O(n)$ après chaque déplacement. Au total on a $O(nm)$.
5. Par l'absurde, supposons que v est tel que $\deg_A(v) > \deg_{\overline{A}}(v)$. On montre que dans ce cas v est déplaçable : en effet, si on fait le déplacement, on gagne $\deg_A(v)$ arêtes dans la coupe, et on perd $\deg_{\overline{A}}(v)$ dans la coupe, et comme $\deg_A(v) > \deg_{\overline{A}}(v)$, la coupe a grandi. Mais alors cela ne peut pas être la coupe renvoyée par l'algorithme. Contradiction. Donc $\deg_A(v) \leq \deg_{\overline{A}}(v)$.
6. Le membre de gauche compte pour chaque sommet de A, le nombre d'arête allant vers un autre sommet de A. Quand on fait cette somme, on ne s'intéresse qu'aux arêtes avec les deux extrémités dans A (celles de X). Pour chaque telle arête, on la visite exactement deux fois : une fois pour chaque extrémité. On obtient alors $2|X|$.
7. On prend l'inégalité de la question 5, $\deg_A(v) \leq \deg_{\overline{A}}(v)$, et on somme sur les sommet de A :

$$\sum_{v\in A} \deg_A(v) \leq \sum_{v\in A} \deg_{\overline{A}}(v)$$

   Le membre de gauche est égale à $2|X|$ d'après la question 6. Dans le membre de droite, pour chaque sommet de A, on compte le nombre d'arêtes allant vers $\overline{A}$,

ce qui revient à compter le nombre d'arêtes de la coupe. Ce nombre est $|Z|$, par définition de Z. Ce qui montre la première inégalité. La deuxième inégalité est juste le symétrique de celle-ci, en inversant A et $\overline{A}$.

8. On a $m = |X| + |Y| + |Z|$ car $X \cup Y \cup Z$ forment une partition des arêtes. Donc $m = |X| + |Y| + |Z| \leq \frac{1}{2}|Z| + \frac{1}{2}|Z| + |Z|$, c'est-à-dire $m \leq 2|Z|$.
9. Comme la solution optimale est au plus égale à m, on a une 2-approximation.

Exercice 5

1. Quand $k = 1$, on cherche le point P qui minimise $\max_{1 \leq i \leq n} d(P, P_i)$. Appelons cette quantité le *rayon* de P. Calculer le rayon prend un temps $O(n)$ (c'est un max sur n valeurs). Il suffit donc de calculer les rayons de tous les points (en $O(n^2)$), et de choisir celui de plus petit rayon.
2. $P_1 = A$. Le point le plus loin de A est C, d'où $P_2 = C$. Il faut maintenant trouver le point le plus loin de $\{A, C\}$. Les distances des autres points à $\{A, C\}$ sont :

	B	D	E	F
AC	1	3	2	1

   Le point le plus loin est D. l'ensemble obtenu est donc $\{A, C, D\}$, et le point le plus loin de $\{A, C, D\}$ est E, qui est à distance 2. La valeur demandée est donc 2.
3. Il s'agit d'un algorithme glouton.
4. Pour trouver P_i à partir des points précédents, on doit trouver le point le plus loin de $\{P_1, \ldots, P_{i-1}\}$. Pour calculer la distance d'un point à $\{P_1, \ldots, P_{i-1}\}$, on fait un min de $i-1$ valeurs, donc on utilise $O(i) = O(k)$ opérations. Pour obtenir P_i, on cherche parmi n points celui ayant la plus grande distance, ce qui coûte donc $n \times O(k) = O(nk)$.

   Comme on itère ce processus k fois, la complexité finale est $O(nk^2)$.
5. (a) Sans perte de généralité, $i < j$. Au moment de choisir P_j, l'algorithme a pris le point le plus loin de $\{P_1, \ldots, P_{j-1}\}$. Il existe P à distance au moins $> 2d^*$, donc le P_j choisi est au moins à distance $> 2d^*$ de $\{P_1, \ldots, P_{j-1}\}$, et donc en particulier de P_i.

   (b) On commence par l'unicité : s'il y a deux points P_j et $P_{j'}$ à distance $\leq d^*$ de P_i^*, alors P_j et $P_{j'}$ se trouvent dans le disque de centre P_i^* et de rayon d^*, c'est-à-dire un cercle de diamètre $2d^*$. Or, P_j et $P_{j'}$ sont à distance $> 2d^*$ l'un de l'autre, ce qui les empêche d'être tous les deux dans le cercle.

   Pour l'existence, puisque chaque point P_j ne peut être proche que d'un seul P_i^*, et qu'il y a autant de P_j que de P_i^*, chaque P_j doit être associé à au moins un P_i^* (sinon, deux P_j seront associés au même P_i^*).

   (c) C'est la définition du coût d^* de la solution optimale.

   (d) P est à distance $\leq d^*$ d'un P_i^* qui est lui-même à distance $\leq d^*$ d'un P_j. Donc P est à distance $\leq 2d^*$ de ce P_j par inégalité triangulaire. Contradiction. Ainsi, on a bien une 2-approximation.