# DS 4 Mathématiques
# Correction

## Exercice 1

*f* Définie si et seulement si $2e^{2x} - 7e^x + 6 > 0$. On pose $y = e^x$, alors cette équation est équivalente à $2y^2 - 7y + 6 > 0$. C'est un polynôme de degré 2, avec $\Delta = 7^2 - 4 \times 2 \times 6 = 49 - 48 = 1$, qui admet pour solutions $\frac{7\pm1}{2\times2}$ soit $x_1 = \frac{6}{4} = \frac{3}{2}$ et $x_2 = \frac{8}{4} = 2$. Alors $2y^2 - 7y + 6 > 0$ pour $y < \frac{3}{2}$ ou pour $y > 2$. Reste à résoudre : $e^x < \frac{3}{2} \Leftrightarrow x < \ln\left(\frac{3}{2}\right)$ et $e^x > 2 \Leftrightarrow x > \ln(2)$. En conclusion le domaine de définition est

$$\mathscr{D}_f = \left]-\infty, \ln\left(\tfrac{3}{2}\right)\right[ \cup \left]\ln(2), +\infty\right[ \tag{1}$$

*g* La fonction racine cubique est définie sur $\mathbb{R}$ (car la fonction $x \mapsto x^3$ est bijective de $\mathbb{R}$ dans $\mathbb{R}$), mais la fonction racine quatrième est seulement définie sur $[0, +\infty[$ tout comme la racine carrée (la fonction $x \mapsto x^4$ est bijective de $[0, +\infty[$ dans lui-même). En conclusion $g$ est définie si et seulement si $x \geqslant 3$, c'est à dire

$$\mathscr{D}_g = [3, +\infty[ \tag{2}$$

*h* Pour définir $h$, il faut $x \neq 0$ (pour le $\frac{1}{x}$) et aussi $\lfloor \frac{1}{x} \rfloor \neq 0$. Or par définition

$$\left\lfloor \frac{1}{x} \right\rfloor = 0 \iff 0 \leqslant \frac{1}{x} < 1 \tag{3}$$

$$\iff 0 < \frac{1}{x} < 1 \tag{4}$$

$$\iff x > 1 \tag{5}$$

$$\tag{6}$$

Donc $h$ est définie si et seulement si $x \leqslant 1$ et $x \neq 0$. En conclusion

$$\mathscr{D}_h = ]-\infty, 0[ \cup ]0, 1] \tag{7}$$

*i* Pour que $i$ soit définie, il faut que $\tan(3x)$ soit définie, que $\arctan(3x)$ soit défini et que $\arctan(3x) \neq 0$.
Or d'une part $\tan(3x)$ n'est pas définie si et seulement si $\exists k \in \mathbb{Z}$, $3x = \frac{\pi}{2} + k\pi$ c'est à dire $x = \frac{\pi}{6} + \frac{k\pi}{3}$.
D'autre part $\arctan(3x) = 0 \Leftrightarrow 3x = 0 \Leftrightarrow x = 0$ (et ce 0 n'est pas un $\frac{\pi}{6} + \frac{k\pi}{3}$).
En résumé

$$\mathscr{D}_i = \mathbb{R} \setminus \left( \left\{ \frac{\pi}{6} + \frac{k\pi}{3} \;\middle|\; k \in \mathbb{Z} \right\} \cup \{0\} \right) \tag{8}$$

## Exercice 2

1. Il s'agit de placer 47 personnes sur 53 places. C'est typiquement un nombre d'arrangements (injection de l'ensemble des personnes vers l'ensemble des places), le premier choisi sa place parmi 53, le deuxième parmi 52, etc et le dernier la choisit parmi 7 places, à la fin 6 places sont libres. Bref c'est $\frac{53!}{(53-47)!} = \frac{53!}{6!} = 53 \times \cdots \times 7$.
2. Le problème est analogue à celui des groupes de colle. D'abord on numérote les groupes. Il y a $\binom{45}{2}$ choix pour le premier binôme, puis $\binom{43}{2}$ choix pour le second etc, et à la fin il reste 5 personnes qui forment le dernier groupe. Le nombre de possibilités s'écrit alors $\frac{45\times44}{2} \times \frac{43\times42}{2} \times \cdots \times \frac{5\times4\times3\times2\times1}{5\times4\times3\times2\times1}$ qu'on peut aussi écrire $\frac{45!}{2^{20}\times5!}$. Comme l'ordre des groupes n'a pas d'importance, il faut diviser ceci par le nombre de permutations, qui est ici… 20! (le 21-ème groupe, celui à 5 personnes, reste à part !) chaque configuration étant comptée avec tous les ordres possibles des binômes.

Dans les questions suivantes, il faut asseoir 45 élèves sur les 49 places hors de la première rangée, et 2 accompagnateurs sur les 4 premières places. Les accompagnateurs ont $4 \times 3$ façons de choisir leur places (nombre d'arrangement de 2 personnes sur 4 places), les réponses seront donc toutes multiples de $4 \times 3$…

3. Pour asseoir 45 élèves sur 49 places il y a donc $\frac{49!}{(49-45)!} = \frac{49!}{4!}$ possibilités. Comme dit ci-dessus, ceci est à multiplier par les possibilités d'asseoir les organisateurs. Le résultat final est $\frac{49!}{4!} \times 4 \times 3$.

4. (a) On choisit d'abord les 5 élèves qui vont aller au fond, et leur place. Le nombre de possibilités est $45 \times 44 \times 43 \times 42 \times 41$ (quel élève va sur la première des 5 places au fond, puis sur la deuxième place, etc). Ensuite il faut installer 40 élèves sur 44 places restantes (le bus sans la rangée du fond ni la première rangée), le résultat est le nombre d'arrangements $\frac{44!}{4!}$. Sans oublier les accompagnateurs à la fin : le résultat final est $45 \times 44 \times 43 \times 42 \times 41 \times \frac{44!}{4!} \times 4 \times 3$.
   Autre point de vue : on range les élèves en ordre et on les installe, en commençant par remplir les places au fond. Les 5 premiers élèves donnent un nombre de possibilités $5 \times \times 4 \times 3 \times 2 \times 1$, puis installer les autres donne encore $\frac{44!}{4!}$. Mais il faut multiplier tout ceci par le nombre de façon de choisir les 5 premiers élèves qu'on va installer, qui est $\binom{45}{5}$, et on retrouve bien le résultat précédent.
   (b) Rien n'empêche de décider qu'on installe d'abord $X$, parmi les 24 places côtés fenêtres (11 dans la rangée gauche mais sans les places des accompagnateurs, idem à droite, et dans la rangée du fond il y a aussi une fenêtre de chaque côté). Puis on installe 44 élèves sur 48 places restantes. Le résultat est $24 \times \frac{48!}{4!}$, à multiplier encore une fois pour les accompagnateurs : $24 \times \frac{48!}{4!} \times 4 \times 3$.
   (c) On peut décider d'installer d'abord $X$ puis $Y$. Le premier a 49 places possibles, le second en a 47 (si $X$ s'assoit sur une des 44 places dans une rangée) et 44 si $X$ s'assoit sur une des 5 places au fond. Le nombre de possibilités de placer seulement $X$ puis $Y$ est donc $44 \times 47 + 5 \times 44$. Il faudra diviser cela par 2 car cela donne la même configuration si $Y$ choisit sa place avant $X$. Ensuite on installe les 43 élèves restants sur 47 places, puis les accompagnateurs. Bref, le résultat est $\frac{44\times47+5\times44}{2} \times \frac{47!}{4!} \times 4 \times 3$.

## Problème 1

1. (a) $\arccos(x)$ est défini si et seulement si $x \in [-1,1]$ (c'est l'unique $\theta \in [0,\pi]$ tel que $\cos(\theta) = x$, mais $\cos(\theta)$ prend ses valeurs dans $[-1,1]$). C'est aussi $\mathscr{D}_1$, car il s'agit simplement d'être capable de calculer $\arccos(u_0)$.
   (b) Pour calculer $u_2$ il faut d'abord calculer $u_1$ (donc $u_0 \in [-1,1]$) puis calculer $\arccos(u_1)$. On doit donc avoir $\arccos(u_0) \in [-1,1]$. La fonction arccos étant décroissante avec $\arccos(-1) = \pi$ et $\arccos(1) = 0$ (et $\pi < 1$), cette équation est équivalente à $\arccos(u_0) \in [0,1]$ puis $u_0 \in [\cos(1), \cos(0)]$. Ceci est $\mathscr{D}_2 = [\cos(1), 1]$.

| $x$ | $-1$ | | $0$ | | $\cos(1)$ | | $1$ |
|---|---|---|---|---|---|---|---|
| $\arccos(x)$ | $\pi$ | ↘ | $\frac{\pi}{2}$ | ↘ | $1$ | ↘ | $0$ |

(9)

   (c) Poursuivons : pour calculer $u_3$ il faut être capable de calculer $u_1$ puis $u_2$ (donc déjà $u_0 \in \mathscr{D}_2$) et en plus de calculer $\arccos(u_2)$. Il faut donc $u_2 \in [-1,1]$, c'est à dire $\arccos(u_1) \in [-1,1]$ soit (comme précédemment) $u_1 \in [\cos(1),1]$. Mais alors on est ramené à déterminer les $u_0$ tels que $\arccos(u_0) \in [\cos(1),1]$. Là encore cela signifie $u_0 \in [\cos(1), \cos(\cos(1))] = \mathscr{D}_3$.
2. (a) Le résultat se montre par récurrence sur $n \geqslant 1$.
   Pour $n = 1$ : c'est simplement l'affirmation que $u_1 = \arccos(u_0)$.
   Hérédité : soit $n \geqslant 1$, supposons $u_n = f^n(u_0)$. Alors par définition $u_{n+1} = \arccos(f^n(u_0))$. Mais par associativité de la composition, ceci est

$$(\arccos \circ (\underbrace{\arccos \circ \cdots \circ \arccos}_{n \text{ fois}}))(u_0) \tag{10}$$
$$= (\underbrace{\arccos \circ \cdots \circ \arccos}_{n+1 \text{ fois}})(u_0) \tag{11}$$
$$= f^{n+1}(u_0) \tag{12}$$

   et ceci démontre la propriété au rang $n+1$.
   Conclusion : $\forall n \geqslant 1$, $u_n = f^n(u_0)$.
   *Remarque* 1. Ceci est vrai en toute généralité pour une suite vérifiant une relation de récurrence $u_{n+1} = f(u_n)$, alors $u_n = f^n(u_0)$. Il est très cohérent de définir $f^0$ comme étant la fonction identité, ainsi $f^0(u_0) = u_0$. Pour éviter la confusion avec les puissances, on note aussi cela $f^{\circ n}$.

(b) Démontrons séparément les deux inclusions pour éviter les confusions.
Inclusion $\subset$ : soit un nombre $u_0 \in \mathscr{D}_{n+1}$, cela signifie qu'on peut définir tout les termes de la suite jusqu'à $u_{n+1}$. L'élément $u_0$ s'écrit de façon unique $u_0 = \cos(\theta)$ avec $\theta \in [0, \pi]$, c'est $\theta = \arccos(u_0)$. Vérifions alors que $\theta \in \mathscr{D}_n$. Par définition $\arccos(\theta) = u_0$, et donc

$$f^n(\theta) = (\underbrace{\arccos \circ \cdots \circ \arccos}_{n \text{ fois}})(\theta) \tag{13}$$
$$= (\underbrace{\arccos \circ \cdots \circ \arccos}_{n \text{ fois}})(\arccos(u_0)) \tag{14}$$
$$= (\underbrace{\arccos \circ \cdots \circ \arccos}_{n+1 \text{ fois}})(u_0) \tag{15}$$

et ceci est bien défini car $u_0 \in \mathscr{D}_{n+1}$. Ceci démontre l'inclusion $\mathscr{D}_{n+1} \subset \cos(\mathscr{D}_n)$.
Inclusion $\supset$ : de même, soit un élément $u_0 \in \cos(\mathscr{D}_n)$, alors on peut écrire $u_0 = \cos(\theta)$ avec $\theta in \mathscr{D}_n$. Les même calculs que ci-dessus montrent que $f^{n+1}(u_0) = f^n(\theta)$ et donc ceci est bien défini car $\theta \in \mathscr{D}_n$.
En conclusion on a bien l'égalité d'ensembles $\mathscr{D}_{n+1} \subset \cos(\mathscr{D}_n)$.

(c) On en déduit alors $\mathscr{D}_4 = \cos(\mathscr{D}_3) = \cos\Big([\cos(1), \cos(\cos(1))]\Big)$. Mais comme la fonction cos est décroissante (on est très certainement toujours sur $[0, \pi]$) cette image directe ne peut être que $\mathscr{D}_4 = [\cos(\cos(1)), \cos(\cos(\cos(1)))]$.
De même, $\mathscr{D}_5 = \cos(\mathscr{D}_4) = \cos\Big([\cos(\cos(1)), \cos(\cos(\cos(1)))]\Big) = [\cos(\cos(\cos(\cos(1)))), \cos(\cos(\cos(1)))]$.

3. (a) On note d'un seul coup la propriété de récurrence $\mathcal{P}(k)$ : « $0 \leqslant c_{2k+1} \leqslant c_{2k+3} \leqslant c_{2k+2} \leqslant c_{2k} \leqslant 1$ » pour $k \in \mathbb{N}$.
$k = 0$ : il s'agit de démontrer $0 \leqslant c_1 \leqslant c_3 \leqslant c_2 \leqslant 1$. On sait que $c_0 = 1$. On en déduit, sachant que cos est décroissante sur $[0, \pi]$ et que $0 < 1 < \frac{\pi}{2}$ que $0 \leqslant \cos(1) \leqslant 1$, ce qui est $0 \leqslant c_1 \leqslant 1$. Encore une fois par décroissance (avec $[0, 1] \subset [0, \pi]$) on en déduit $\cos(1) \leqslant \cos(\cos(1)) \leqslant 1$, ce qui est $c_1 \leqslant c_2 \leqslant 1$. Et encore par décroissance on obtient $\cos(1) \leqslant \cos(\cos(\cos(1))) \leqslant \cos(\cos(1))$ ce qui s'écrit $c_1 \leqslant c_3 \leqslant c_2$. Mises bout à bout, avec ces inégalités on a bien démontré le cas $k = 0$.
Hérédité : soit $k \in \mathbb{N}$, supposons $\mathcal{P}(k)$. De même par décroissance on obtient d'abord

$$\cos(1) \leqslant \cos(c_{2k}) \leqslant \cos(c_{2k+2}) \leqslant \cos(c_{2k+3}) \leqslant \cos(c_{2k+1}) \leqslant \cos(0) \tag{16}$$

c'est à dire $\cos(1) \leqslant c_{2k+1} \leqslant c_{2k+3} \leqslant c_{2k+4} \leqslant c_{2k+2} \leqslant 1$, puis encore en appliquant cos sur tout cela on trouve

$$\cos(1) \leqslant \cos(c_{2k+2}) \leqslant \cos(c_{2k+4}) \leqslant \cos(c_{2k+3}) \leqslant \cos(c_{2k+1}) \leqslant \cos(0) \tag{17}$$

autrement dit $\cos(1) \leqslant c_{2k+3} \leqslant c_{2k+5} \leqslant c_{2k+4} \leqslant c_{2k} \leqslant 1$, et c'est bien la propriété $\mathcal{P}(k+1)$ (il reste seulement à minorer à gauche $\cos(1)$ par 0).

(b) Il s'agit de montrer le résultat par récurrence, en notant $\mathcal{P}(n)$ la propriété proposée pour $n \geqslant 2$.
$n = 2$ : on a déjà vu $\mathcal{D}_2 = [\cos(1), 1]$, et ceci est $[c_1, c_0]$ (avec 2 qui est pair). C'est bien $\mathcal{P}(2)$.
Hérédité : soit $n \geqslant 2$, supposons $\mathcal{P}(n)$. Il faut traiter les deux cas.
Si $n$ est pair, alors par hypothèse $\mathscr{D}_n = [c_{n-1}, c_{n-2}]$ et de plus $n+1$ est impair. Or $\mathscr{D}_{n+1} = \cos(\mathscr{D}_n)$ est (toujours la même histoire : on est de toute façon sur $[0, \pi]$ où cos est continue, décroissante) $[\cos(c_{n-2}), \cos(c_{n-1})]$ c'est à dire $[c_{n-1}, c_n]$. Mais ceci vérifie bien $\mathcal{P}n+1$, puisque $n+1$ est impair.
Si $n$ est impair alors de même, par hypothèse $\mathscr{D}_n = [c_{n-2}, c_{n-1}]$ et donc $\cos(\mathscr{D}_n) = [\cos(c_{n-1}), \cos(c_{n-2})]$ et ceci est $[c_n, c_{n-1}]$, et cela vérifie bien $\mathcal{P}(n+1)$ car $n+1$ est pair.

4. (a) La suite $(c_n)_{n\in\mathbb{N}}$ est à valeurs dans $[0, 1]$, les $(c_{2k})_{k\in\mathbb{N}}$ et les $(c_{2k+1})_{k\in\mathbb{N}}$ sont donc à la fois majorés et minorés. En particulier $\mathscr{A}$ admet bien une borne inférieure $a$ (la suite est décroissante, $a \geqslant 0$) et $\mathscr{B}$ admet bien une borne supérieure $b$ (la suite est croissante, $b \leqslant 1$). De plus tous les termes de $a$ sont supérieurs à tous les termes de $b$ donc on peut montrer que $b \leqslant a$.

(b) Par définition $\cos(\mathscr{A})$ est exactement égal à l'ensemble $\mathscr{B}$. Mais la fonction cos étant décroissante, le cos de la borne inférieur de $a$ est égal à la borne supérieure de $b$, c'est à dire $\cos(a) = b$. En effet $a$ est le plus grand des majorants de $\mathscr{A}$ et donc $\cos(a)$ sera automatiquement le plus petit des minorants de $\cos(\mathscr{A})$.

Presque de même ensuite, car $\cos(\mathscr{B})$ n'est pas exactement égal à l'ensemble $\mathscr{A}$ : c'est l'ensemble $\{c_{2k+2} \mid k \in \mathbb{N}\}$ c'est à dire en fait $\mathscr{A} \setminus \{c_0\}$ avec $c_0 = 1$. On en déduit facilement que $\cos(b)$ est égal à la borne inférieure de $\mathscr{A} \setminus \{1\}$. Or 1 n'est clairement pas le plus petit élément de $\mathscr{A}$ (puisque ce sont les termes d'une suite décroissante) et donc la borne inférieur de cet ensemble est la même que celle de $\mathscr{A}$ (pour les deux, 1 n'est pas dans l'ensemble des minorants, donc le plus grand minorant est le même). On en déduit donc $\cos(b) = a$.

(c) On pose $g : x \mapsto \cos(x) - x$, alors $g$ est dérivable est $\forall x \in [0, \pi]$, $g'(x) = -\sin(x) - 1 \leqslant 0$, donc $g$ est décroissante. De plus $g(0) = 1$ et $g(\pi) = -1 - \pi < 0$.

| $x$ | $0$ | | $c$ | | $\pi$ |
|---|---|---|---|---|---|
| $g'(x)$ | | $+$ | | | |
| $g(x)$ | $1$ | ↗ | $0$ | ↘ | $-1-\pi$ |

(18)

(d) On en déduit que $g$ réalise une bijection de $[0, \pi]$ dans $[-1 - \pi, 1]$, en particulier l'équation $g(x) = 0$ admet une unique solution $c$ tel que $\cos(c) = c$.

Le but est ensuite de montrer que $c = a = b$, partant des égalités $a = \cos(b)$ et $b = \cos(a)$. On déduit de ces égalités $a - b = \cos(b) - \cos(a)$, ce qui s'écrit aussi $a + \cos(a) = b + \cos(b)$. Mais un tableau de variations pour la fonction cette fois $h : x \mapsto x + \cos(x)$ montre que $h$ est strictement croissante, donc injective, et donc cette égalité implique $a = b$.

5. (a) Par définition $\mathscr{D}_\infty$ désigne les $u_0$ qui sont dans tous les $\mathscr{D}_n$, $n \geqslant 1$. Or de part le calcul des $\mathscr{D}_n$ en fonction de $n$, on en déduit que $u_0$ doit être plus petit que tous les $c_{2k}$ ($k \in \mathbb{N}$) (car ce sont les bornes du dessus des intervalles $\mathscr{D}_n$), mais donc doit être inférieur à la borne inférieur de $\mathscr{A}$ c'est à dire $a$. De même, $u_0$ doit être plus grand que tous les $c_{2k+1}$ car ce sont les bornes du dessous des intervalles $\mathscr{D}_n$, et donc doit être plus grand que $b$. Mais comme $a = b$ on en déduit que nécessairement $u_0 = a = b$, et donc $\mathscr{D}_\infty = \{a\} = \{b\}$.

(b) Si la suite $u_n$ est bien définie alors $u_0 \in \mathscr{D}_\infty$ donc $u_0 = a$, l'unique valeur telle que $\cos(a) = a$ (et donc aussi $\arccos(a) = a$). Mais alors on en déduit immédiatement $u_1 = a$ et par récurrence $\forall n \in \mathbb{N}$, $u_n = a$.

## Problème d'informatique

1. Il s'agit dans cette question de parcourir la chaine de caractères `alphabet` et de trouver laquelle des lettres est égale à `x` (on suppose qu'il y en a bien une et une seule, pas la peine de chercher à traiter ce qui se passerait si ce n'est pas le cas).

```
def numéro_lettre(x):
    for j in range(26):
        if alphabet[j] == x:
            return j
```

2. Cette fois-ci il faut compter. On a besoin de savoir à quelle lettre le numéro `j` correspond.

```
def compte_lettre(m, j):
    c = 0
    for i in range(len(m)):
        if m[i] == alphabet[j]:
            c = c + 1
    return c
```

La ligne `if m[i] == alphabet[j]` peut tout aussi bien être remplacée par `if numéro_lettre(m[i]) == j` (il y a là deux bijections réciproques, entre les nombres de $[\![0, 25]\!]$ et les lettres de l'alphabet !)

3. On peut facilement le faire à l'aide de la fonction précédente, en comptant séparément chacune des lettres.

```
def compte_tout(m):
    C = [0] * 26
    for j in range(26):
        C[j] = compte_lettre(m, j)
    return C
```

Une manière plus élégante est d'itérer une seule fois sur `m`, et de chercher pour quelle lettre il faut incrémenter le compteur (remarquez qu'on note par `i` les indices de `m` et par `j` les indices dans `C` c'est à dire les lettres de l'alphabet).

```
def compte_tout(m):
    C = [0] * 26
    for i in range(len(m)):
        j = numéro_lettre(m[i])
        C[j] = C[j] + 1
    return C
```

4. Standard et sans commentaires ! Cela traduit le plus directement possible la définition avec $0! = 1$ et $n! = n \times (n-1)!$.

```
def factoriel(n):
    if n == 0:
        return 1
    else:
        return n * factoriel(n-1)
```

5. On rappelle que pour calculer le nombre d'anagrammes d'un mot, on divise le nombre total de permutations des lettres (c'est $n!$, si on note $n$ la longueur du mot) par le produit des factoriels du nombre de fois où chaque lettre apparait. Le numérateur est donc sans hésiter `factoriel(n)`. Pour le dénominateur, il faut calculer un produit, en itérant sur la liste `C` de longueur 26. En fait, comme $0! = 1$ et $1! = 1$, il n'est pas dérangeant d'inclure dans ce produit les lettres qui n'apparaissent qu'une seule fois et celles qui n'apparaissent pas du tout. Le produit est toujours initialisé à 1.

```
def nombre_anagrammes(m):
    n = len(m)
    C = compte_tout(m)
    numérateur = factoriel(n)
    dénominateur = 1
    for j in range(26):
        dénominateur = dénominateur * factoriel(C[j])
    return numérateur // dénominateur # division en nombres entiers
```

Si on a peur des lettres qui n'apparaissent pas ou bien une seule fois, il suffit de rajouter dans la boucle

```
if C[j] >= 2:
    dénominateur = dénominateur * factoriel(C[j])
```

6. La boucle présentée itère sur la variable `C`, qui compte combien de fois chaque lettre apparait. Elle renvoie `False` (et s'arrête donc) dès que la condition suivante est vérifiée : `C[j] != 0 and C[j] != 1`. Ceci signifie que la $j$-ème lettre de l'alphabet n'apparait pas 0 fois et n'apparait pas 1 fois, le contraire est que la lettre apparait au moins 2 fois. Ainsi, si une lettre apparait au moins 2 fois dans `m` alors la fonction renvoie `False`. Sinon, on arrive au bout de la boucle sans avoir rencontré cette condition et la fonction renvoie `True`.

   En conclusion on peut dire que la fonction teste si chaque lettre apparait au plus une fois. Cela correspond aux mots de longueur $n$ pour lesquels le nombre d'anagrammes est exactement $n!$, sans avoir besoin de diviser.

## Problème 2

1. (a) Les partitions sont simplement

$$3 = 3 \tag{19}$$
$$= 1 + 2 \tag{20}$$
$$= 1 + 1 + 1 \tag{21}$$

   (b) Les partitions de 5 sont :

$$5 = 5 \tag{22}$$
$$= 1 + 4 \tag{23}$$
$$= 2 + 3 \tag{24}$$
$$= 1 + 1 + 3 \tag{25}$$
$$= 1 + 2 + 2 \tag{26}$$
$$= 1 + 1 + 1 + 2 \tag{27}$$
$$= 1 + 1 + 1 + 1 + 1 \tag{28}$$

   On en trouve bien 7. La première ici est de longueur 1, les deux suivantes de longueur 2, les deux d'après de longueur 3, puis une de longueur 4 et une de longueur 5.

   (c) On trouve $p_6 = 11$ en listant les partitions. Comme précédemment, un ordre logique est de les lister récursivement en fonction du maximum qui apparait dans la partition (après $6 = 6$ et $6 = 1+5$, d'abord celles contenant 4 en commençant par $6 = 2 + 4$, puis celles contenant au plus 3, etc).

$$6 = 6 \tag{29}$$
$$= 1 + 5 \tag{30}$$
$$= 2 + 4 \tag{31}$$
$$= 1 + 1 + 4 \tag{32}$$
$$= 3 + 3 \tag{33}$$
$$= 1 + 2 + 3 \tag{34}$$
$$= 1 + 1 + 1 + 3 \tag{35}$$
$$= 2 + 2 + 2 \tag{36}$$
$$= 1 + 1 + 2 + 2 \tag{37}$$
$$= 1 + 1 + 1 + 1 + 2 \tag{38}$$
$$= 1 + 1 + 1 + 1 + 1 + 1 \tag{39}$$

   (d) $p(n, 1) = 1$ correspondant à la seule partition $n = n$, et $p(n, n) = 1$ correspondant à la seule partition $n = 1 + \cdots + 1$. Si $k > n$ alors $p(n, k) = 0$, cela provient du fait que les nombres dans la partition sont au moins égaux à 1 et donc la somme d'une partition de longueur $n$ est au moins égale à $n$.

2. Supposons $n$ pair : il existe $k$ tel que $n = 2k$, et nécessairement $k \geqslant 1$. On a alors une partition évidente de longueur 2 de $n$, c'est $n = k + k$. Mais on peut aussi faire $n = (k - 1) + (k + 1)$, ou $n = (k - 2) + (k + 2)$, etc jusqu'à $n = 1 + (k + k - 1)$. En fait, les partitions de $n$ sont en bijection avec $[\![0, k - 1]\!]$ : étant donné $i \in [\![0, k - 1]\!]$ on forme la partition $(n - i, n + i)$, et réciproquement toute partition de longueur 2 est donnée par un couple $(a, b)$ tels que $a + b = n$ et $1 \leqslant a \leqslant b$, ce qui implique nécessairement $a \leqslant k$ (donc qu'on peut écrire $a = n - i$ avec $i \in [\![0, k - 1]\!]$). Il y a donc exactement $k$ partitions de $n$ de longueur 2.
   Si $n$ est impair, écrivons $n = 2k + 1$, pour $k \geqslant 0$. Alors il y a la partition évidente $n = k + (k + 1)$. Ici il faut séparer le cas $k = 0$ (soit $n = 1$) pour lequel ceci n'est pas une partition : le nombre 1 admet une seule partition $1 = 1$ et celle-ci n'est pas de longueur 2. Sinon, avec le même raisonnement, de $n = k + (k + 1)$ on peut aussi former $n = (k - 1) + (k + 2)$ etc jusqu'à $n = 1 + (k + k)$ et y a $k$ partitions de longueur 2 de $n$.

3. (a) Il y a en réalité peu à dire. Chaque partition dans $E_n$ admet une certaine longueur $k$, avec nécessairement $1 \leqslant k \leqslant n$, elle est donc dans $E_{n,k}$. Et elle est dans un seul à la fois, donc ces ensembles sont deux à deux disjoints.

(b) On obtient alors, par la formule de dénombrement

$$\mathrm{Card}(E_n) = \sum_{k=1}^{n} \mathrm{Card}(E_{n,k}) \tag{40}$$

Mais par définition $\mathrm{Card}(E_n) = p_n$ et $\mathrm{Card}(E_{n,k}) = p(n,k)$.

*Remarque* 2. C'est très formel et en même temps facile : cela sert uniquement à ramener à nos constructions mathématiques habituelles l'idée simple selon laquelle, pour obtenir toutes les partitions de $n$, on somme selon les différents cas selon leur longueur.

4. (a) Là encore c'est très formel mais il y a peu à dire : pour la partition $(a_1, \dots, a_k)$ de longueur $k$ (avec $k \leqslant 1$) alors les coefficients sont au moins 1 et par ordre croissant, donc soi $a_1 = 1$ soit $a_1 > 1$. Cela signifie qu'une partition de $E_{n,k}$ est soit dans $\mathscr{A}$ soit dans $\mathscr{B}$, et jamais les deux en même temps.

(b) On peut traiter cette question avant ou après la suivante. La bijection $\varphi$ est celle que « enlève le premier 1 », c'est à dire ici l'application

$$\begin{aligned} \varphi : \mathscr{A} &\longrightarrow E_{n-1,k-1} \\ (a_1, \dots, a_k) &\longmapsto (a_2, \dots, a_k) \end{aligned} \tag{41}$$

Il faut remarquer que tout est bien défini : $k > 1$ donc $(a_2, \dots, a_k)$ est bien de longueur au moins 1, et de somme $n-1$ ; ses éléments restent non-nuls est rangés par ordre croissants. La bijection réciproque est celle qui « remet 1 » :

$$\begin{aligned} \varphi^{-1} : E_{n-1,k-1} &\longrightarrow \mathscr{A} \\ (a_2, \dots, a_k) &\longmapsto (1, a_2, \dots, a_k) \end{aligned} \tag{42}$$

Rajouter un 1 devant une partition de $n-1$ de longueur $k-1$ donne bien une partition de $n$ de longueur $k$, et cela ne change pas la condition que les éléments sont rangés par ordre croissant puisque 1 est le minimum possible.

(c) Il faut vérifier que tout est bien défini. La partition $(a_1 - 1, \dots, a_k - 1)$ est bien de même longueur $k$, ses éléments sont toujours rangés par ordre croissant, la somme est cette fois-ci $\sum_{i=1}^{k}(a_i - 1) = \sum_i^k a_i - k = n - k$. Il faut remarque que comme au départ $a_1 > 1$ alors tous les $a_i$ sont $> 1$ et donc $a_i - 1 \geqslant 1$, donc on a bien une partition de longueur $k$ de $n - k$.

La bijection réciproque est celle qui ajoute 1 à chaque élément de la partition :

$$\begin{aligned} \psi^{-1} : E_{n-k,k} &\longrightarrow \mathscr{B} \\ (b_1, \dots, b_k) &\longmapsto (b_1 + 1, \dots, b_k + 1) \end{aligned} \tag{43}$$

Partant d'une partition de $n - k$, cela donne bien entendu une partition de $n$. De plus sachant $b_1 \geqslant 1$ alors $b_1 + 1 > 1$, l'image est bien dans $\mathscr{B}$. Montrer que l'application $\psi$ qui à une partition $(a_1, \dots, a_k)$ de $n$ associe la partition $(a_1 - 1, \dots, a_k - 1)$ est une bijection entre $\mathscr{B}$ et $E_{n-k,k}$.

(d) On en déduit alors par la première question

$$\mathrm{Card}(E_{n,k}) = \mathrm{Card}(\mathscr{A}) + \mathrm{Card}(\mathscr{B}) \tag{44}$$

puis avec les bijections qu'on a exhibées

$$\mathrm{Card}(\mathscr{A}) = \mathrm{Card}(E_{n-1,k-1}) + \mathrm{Card}(E_{n-k,k}) \tag{45}$$

Cela donne la relation cherchée $p(n,k) = p(n-1,k-1) + p(n-k,k)$

5. Les questions précédentes nous donnent toutes les valeurs de $p(n,k)$ pour $1 \leqslant n \leqslant 6$, ainsi que les bords du tableau. On peut alors compléter le reste du tableau à l'aide de la relation de récurrence établie.

| $n \backslash k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | | | | | | | |
| 2 | 1 | 1 | | | | | | |
| 3 | 1 | 1 | 1 | | | | | |
| 4 | 1 | 2 | 1 | 1 | | | | |
| 5 | 1 | 2 | 2 | 1 | 1 | | | |
| 6 | 1 | 3 | 3 | 2 | 1 | 1 | | |
| 7 | 1 | 3 | 4 | 3 | 2 | 1 | 1 | |
| 8 | 1 | 4 | 5 | 5 | 3 | 2 | 1 | 1 |

(46)

(a) La fonction contiendra bien sûr une ligne telle que `return p(n-1, k-1) + p(n-k, k)`. Mais il faut analyser un peu plus finement la condition à laquelle la récurrence se termine (le cas d'initialisation de la fonction récursive). Ce qui est certain est que $n$ diminue dans la récurrence si on a bien $k > 0$, il faut donc traiter le cas minimal $n = 1$. Mais il faut aussi tenir compte du fait que la relation de récurrence n'est valable que pour $n > k > 1$, autrement dit on doit inclure les cas $k = n$ et $k = 1$ dans l'initialisation. On pourrait penser que ce n'est pas si grave car on remontera de toute façon à $n = 1$ comme pour le calcul des coefficients binomiaux, mais ce n'est pas le cas ici : pour $k = 0$ l'appel à `p(n-k, k)` ne fait pas diminuer $n$ et produit une récurrence infinie ! On doit donc aussi traiter les cas en dehors $k < 1$ ou $k > n$, qui arrivent naturellement dans la récurrence (par exemple $p(7, 4)$ nécessite de calculer $p(3, 4)$, à qui on doit donner la valeur 0). Bref, une façon propre et sécurisée de faire, qui ne produit pas de récurrence infinie pour des valeurs initiales $n \geqslant 1$ et $k \in \mathbb{Z}$ quelconques, est :

```
def p(n, k):
    if n == 1:
        if k == 1:
            return 1
        else:
            return 0
    elif k == 1 or k == n:
        return 1
    elif k < 1 or k > n:
        return 0
    else:
        return p(n-1, k-1) + p(n-k, k)
```

Après coup, on peut enlever le test `n == 1`, car il se retrouve inclus dans les deux autres ; alternativement, on peut enlever le test `k == 1 or k == n`, car effectivement la relation de récurrence pour $k \neq 0$ fera bien remonter jusqu'à $n = 1$.

(b) On peut le faire avec une simple somme.

```
def partitions(n):
    S = 0
    for k in range(1, n+1):
        S = S + p(n, k)
    return S
```

*Remarque* 3. Pour ces deux fonctions, même si la formulation récursive semblait la plus simple et directe à programmer, c'est loin d'être une méthode efficace (à cause de l'arbre des appels, comme pour Fibonacci ou pour le calcul des coefficients binomiaux dans leur version récursive). La bonne méthode est, comme dans la question précédente, de calculer au fur et à mesure tout le *tableau* (qu'on implémentera en Python comme une liste de listes) des valeurs de $p(n, k)$.