Exercise 13 (non-corrigé)

Par observation, j'obtiens ceci :

soit $x = \frac{n+k}{2}$.

Si $x \notin \mathbb{N}$, alors il n'y a pas de chemin qui mènent à k en n étapes.

Si $x \in \mathbb{N}$, alors il y a $\binom{n}{x}$ chemins qui mènent à k en n étapes.

J'ai déduis ma réponse avec le triangle de Pascal. Essayons de le montrer rigoureusement :

Soit $p \in \mathbb{N}$ le nombre d'étapes $+1$ et $q \in \mathbb{N}$ le nombre d'étapes -1 d'un chemin de longeur n qui mène à k. Nous avons $n = p + q$ et $k = p - q$, donc

$$\begin{aligned} & k = p - q \\ \implies \quad & k - p = -q \\ \implies \quad & q = p - k \end{aligned}$$

Nous pouvons trouver la valeur de p en fonction de n et k

$$\begin{aligned} & n = p + (p - k) \\ \implies \quad & n = 2p - k \\ \implies \quad & p = \frac{n+k}{2} \end{aligned}$$

Maintenant que l'on a p, calculons le nombre de chemins que nous pouvons construire avec p étapes $+1$ et q étapes -1. Nous savons qu'il y a n étapes, choisissons p de ces étapes, nous avons $\binom{n}{p}$ possibilités car l'ordres des étapes $+1$ n'est pas important. Maintenant nous avons qu'une seul possibilité pour remplir les q étapesréstantes avec -1, donc nous n'avons pas plus de choix.

Dans les cas où notre calcul mène à $(p, q) \notin \mathbb{N}^2$ nous avons une contradiction et il n'existe pas de chemin de longeur n qui mènent à k.

Exercise 14 (non-corrigé)

Avec le même résonemment utilisé dans l'exercise 13 : tous les chemins qui mènent à (p, q) contiennent p étapes « droite » et q étapes « haut », donc ils ont également $p + q$ étapes.

Nous pouvons conclure qu'il existe $\binom{p+q}{p}$ chemins qui mènet à (p, q).

Exercise 15 (non-corrigé)

Soit E un ensemble de 10 patates tel que la somme de leur masse soit 1000 grammes. Il y a $\text{card}(\mathscr{P}(E)) = 2^{\#E}$ moyens de remplir un sac avec une partie de E, soit 1024 dans notre cas.

Il y à 1001 possibilités de masse du sac (un sac sans masse est possible). Dans le meilleur des cas, nous pouvons avoir 1001 configurations de patates qui mènent à un sac de masse différente, or nous avons un total de 1024 configurations possible.

Nous pouvons conclure qu'il existe au moins 23 configurations qui mènent à un sac de même masse qu'une autre configuration.

Exercise 16 (non-corrigé)

Sois polygone convexe à n sommets.

Il a n côtés.

Il a $n(n-3)$ diagonales.

Ses diagonales ont au maximum $\frac{1}{24}n\left((n-2)^3-(n-2)\right)$ intersection entre elles.

Essayons de le montrer rigoureusement :

Sois un polygone convexe à n sommets. Appelons S l'ensemble de ses sommets.

Sois $s_1 \in S$, s_1 a deux sommets adjacent s_2 et s_n et il forme des diagonales avec chaque $s \in \{s_3, \ldots, s_{n-1}\}$.

Soit $d_k = (s_1, s_{k+2})$, nous avons les sommets à « droite » de la diagonale $S_{\text{droite}} = \{s_2, \ldots, s_{k+1}\}$ et les sommets à « gauche » de la diagonale $S_{\text{gauche}} = \{s_{k+3}, \ldots, s_{n-1}\}$.

Les seuls diagonales qui croisent d appartiennent à $S_{\text{droite}} \times S_{\text{gauche}}$, donc d croise $\#S_{\text{droite}}\#S_{\text{gauche}}$ ou $k(n-2-k)$ diagonales (car le polygone est convexe).

Posons $m = n-2$,

La sommes des intersections qui inclut les diagonales partant de s_1 est donc

$$\sum_{k=1}^{m} k(m-k) = m\sum_{k=1}^{m} k - \sum_{k=1}^{m} k^2$$

appelons la i_1.

Nous savons que

$$\sum_{k=1}^{m} k = \frac{m(m+1)}{2}$$

et que

$$\sum_{k=1}^{m} k^2 = \frac{m(m+1)(2m+1)}{6}$$

donc

$$\begin{aligned} i_1 &= m \cdot \frac{m(m+1)}{2} - \frac{m(m+1)(2m+1)}{6} \\ &= \frac{1}{6}\left(3m^3 + \cancel{3m^2} - 2m^3 - \cancel{3m^2} - m\right) \\ &= \frac{1}{6}\left(m^3 - m\right) \end{aligned}$$

Pour avoir le nombre total d'intesections nous pouvons répéter ce calcul pour chaque sommet, or chaque intersection sera compté quatre fois, une fois pour chaque sommet des deux diagonales.

Nous avons donc $\frac{1}{24}n\left((n-2)^3-(n-2)\right)$ intersections.

Exercise 17 (non-corrigé)

Implémentation en Python3 :

```
import math

def nCr(n: int, r: int) -> int:
    f = math.factorial
    return f(n) // f(r) // f(n - r)

def inputInteger(msg: str) -> int:
    while True:
        n = input(msg)
        if n.isdigit():
            return int(n)

if __name__ == '__main__':
    n = inputInteger('Combiens de marches à l’escalier ? ')
    total = 0
    for n3 in range(n//3 + 1):
        for n2 in range((n - n3*3)//2 + 1):
            n1 = n - n2*2 - n3*3
            m = n1 + n2 + n3
            total += nCr(m, n1)*nCr(m - n1, n2)
    print(f'Il y a {total} façon(s) de monter l’escalier \
en groupes de 1, 2 et 3 marches.')
```