Ch. 3 — Démonstration par récurrence

1 Récurrence simple

Pour montrer qu'une propriété (plus précisément un prédicat) $\mathcal{P}_n$ dépendant d'un entier n est vraie pour tout entier n supérieur ou égal à n_0, il suffit de prouver les deux propriétés suivantes :

(i) la propriété est vraie pour l'entier n_0 ;

(ii) si la propriété est vraie pour l'entier $n \geqslant n_0$, alors elle est vraie pour l'entier $n+1$.

Exemple. Montrons, à l'aide d'un raisonnement par récurrence, que

$$\forall n \in \mathbb{N}^* \qquad 1+2+3+\cdots+n = \frac{n(n+1)}{2}$$

On note $\mathcal{P}_n$: « $1+2+3+\cdots+n = \frac{n(n+1)}{2}$ »

Montrons par récurrence sur n que la propriété $\mathcal{P}_n$ est vraie pour tout entier $n \geqslant 1$.

- *Initialisation de la récurrence.* Montrons que la propriété $\mathcal{P}_1$ est vraie.
  On a
  $$\frac{1(1+1)}{2} = 1$$
  Donc $\mathcal{P}_1$ est vraie.
- *Hérédité.* Supposons la propriété $\mathcal{P}_n$ vraie pour un certain entier $n \geqslant 1$.
  Montrons que la propriété $\mathcal{P}_{n+1}$ est vraie.
  $$\begin{aligned} 1+2+3+\cdots+n+(n+1) &= (1+2+3+\cdots+n)+(n+1) \\ &= \frac{n(n+1)}{2}+(n+1) \quad \text{d'après } \mathcal{P}_n \\ &= (n+1)\left(\frac{n}{2}+1\right) \\ &= (n+1)\frac{n+2}{2} \\ &= \frac{(n+1)((n+1)+1)}{2} \end{aligned}$$
  Donc la propriété $\mathcal{P}_{n+1}$ est vraie.

D'après le principe de récurrence, la propriété $\mathcal{P}_n$ est vraie pour tout entier $n \geqslant 1$.

Exercice nº 1. Démontrer par récurrence que

$$\forall n \in \mathbb{N}^* \qquad 1^2+2^2+3^2+\cdots+n^2 = \frac{n(n+1)(2n+1)}{6}$$

Exercice nº 2. Montrer que, pour tout entier $n \in \mathbb{N}^*$, pour tout réel $q \neq 1$, on a :

$$1+q+q^2+\cdots+q^n = \frac{1-q^{n+1}}{1-q}$$

Exercice nº 3. Montrer que $\quad \forall n \in \mathbb{N}^* \qquad 2^{n-1} \leqslant n!$

Exercice nº 4. *Inégalité de Bernoulli*
Montrer que $\quad \forall x \geqslant -1 \qquad \forall n \in \mathbb{N} \qquad 1+nx \leqslant (1+x)^n \qquad$ avec égalité ssi $x=0$

Remarque. Une démonstration par récurrence peut s'envisager lorsqu'on est en présence d'une conjecture dépendant d'un entier n. La difficulté d'une telle démonstration réside en général dans la nécessité d'avoir formulé une telle conjecture. Souvent, l'étude de la situation pour quelques valeurs particulières et simples de n permet de se faire une idée de la conjecture.

Exercice n° 5. Pour $n \in \mathbb{N}^*$, on pose $u_n = 1^3 + 2^3 + 3^3 + \cdots + n^3 = \sum_{k=1}^{n} k^3$

a) Calculer u_1, u_2, u_3, u_4, u_5.

b) Comparer ces résultats aux valeurs de $1 + 2 + 3 + \cdots + n$ pour $n \in [\![1, 5]\!]$.
Émettre une conjecture concernant l'expression de u_n en fonction de n.

c) Prouver alors cette conjecture à l'aide d'un raisonnement par récurrence.

Exercice n° 6. *Somme télescopique*

a) Justifier la relation $\forall x \in \mathbb{R} \setminus \{0, -1\} \quad \dfrac{1}{x(x+1)} = \dfrac{1}{x} - \dfrac{1}{x+1}$

pour calculer $S_n = \displaystyle\sum_{k=1}^{n} \frac{1}{k(k+1)}$ en fonction de $n \in \mathbb{N}^*$

b) Redémontrer le résultat obtenu par une démonstration par récurrence.

Exercice n° 7. On considère la suite $(v_n)_{n \geqslant 0}$ définie par $\begin{cases} v_0 = 1 \\ \forall n \in \mathbb{N} \qquad v_{n+1} = \sqrt{2 + v_n} \end{cases}$

a) Écrire un programme Python d'entête `suite(n)` qui renvoie la valeur de v_n.
Conjecturer la monotonie de la suite $(v_n)_n$.

b) Montrer par récurrence que $\forall n \in \mathbb{N} \quad 0 < v_n < 2$

c) Étudier le signe du polynôme $P(x) = 2 + x - x^2$

d) Étudier la monotonie de la suite $(v_n)_n$

Exercice n° 8. Pour chacune des suites suivantes, exprimer u_n en fonction de n et déterminer sa limite :

a) $\begin{cases} u_0 = 1 \\ u_{n+1} = \dfrac{u_n}{1 + u_n} \end{cases}$ b) $\begin{cases} u_0 = 1 \\ u_{n+1} = \dfrac{u_n}{\sqrt{1 + u_n^2}} \end{cases}$ c) $\begin{cases} u_0 = 1 \\ u_{n+1} = \dfrac{u_n}{(1 + \sqrt{u_n})^2} \end{cases}$

Exercice n° 9. Montrer que, pour tout $n \in \mathbb{N}$, $3^{n+3} - 4^{4n+2}$ est divisible par 11

2 Récurrence double

Il peut arriver que, pour l'hérédité, quand il s'agit de démontrer $\mathcal{P}_{n+1}$, on ait besoin de supposer la propriété aux deux rangs précédents, c'est-à-dire non seulement pour n, mais aussi pour $n - 1$. On est amené à utiliser le *principe de récurrence double.* On prouve alors que :

(i) la propriété est vraie pour l'entier $n = 0$;
(ii) la propriété est vraie pour l'entier $n = 1$;
(iii) si la propriété est vraie pour l'entier n et pour l'entier $n+1$, alors elle est vraie pour l'entier $n+2$.

Exercice n° 10. *Une relation de récurrence linéaire d'ordre 2*

On définit la suite u par : $\begin{cases} u_0 = 2 \quad u_1 = 5 \\ \forall n \in \mathbb{N} \quad u_{n+2} = 5u_{n+1} - 6u_n \end{cases}$

a) Calculer u_n pour tout $n \in [\![0, 4]\!]$

b) Écrire un programme Python d'entête `suite(n)` qui renvoie la valeur de u_n.

c) Montrer que $\forall n \in \mathbb{N} \quad u_n = 2^n + 3^n$

3 Récurrence forte

Il arrive que pour prouver $\mathcal{P}_{n+1}$, on ait besoin de supposer que $\mathcal{P}_k$ soit vraie pour tout entier $k \leqslant n$. On parle alors de *récurrence forte.* Il suffit de prouver que

(i) la propriété est vraie pour l'entier $n = 0$;
(ii) si la propriété est vraie pour tout entier $k \leqslant n$, alors elle est vraie pour l'entier $n + 1$.

Exercice n° 11. Montrer que tout entier naturel supérieur ou égal à 2 possède un diviseur premier.