

Preuves

- Preuve directe
- Par contre-exemple
- Par induction
- Par contradiction

Preuve directe

Suite de raisonnements qui mènent à l'énoncé

Théorème: la somme de deux entiers pairs est paire

$$(\forall n, m \in \mathbb{N}, n \% 2 = 0 \wedge m \% 2 = 0 \Rightarrow (n+m) \% 2 = 0)$$

Preuve: Soient n et m deux entiers pairs arbitraires

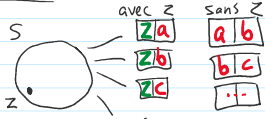
Donc, $n = 2a$ et $m = 2b$ pour des entiers a et b .

On déduit que $n+m = 2a+2b = 2(a+b)$.

Donc $n+m$ est pair. $\square$

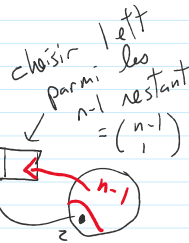
Théorème: $\forall n \geq 2, \binom{n}{2} = \binom{n-1}{1} + \binom{n-1}{2}$ $\binom{n}{2} = |\{X: X \subseteq S \wedge |X|=2 \wedge |S|=n\}|$

Preuve: Soit S un ensemble de taille n .



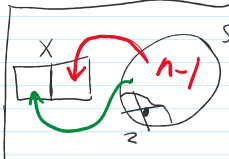
Soit $z \in S$ choisi arbitrairement.

Pour former un sous-ens. $X \subseteq S, |X|=2$ de S , on a 2 cas: soit $z \in X$, ou $z \notin X$.



- Le # de façons d'avoir X avec $z \in X$ est $\binom{n-1}{1}$ car on doit choisir 1 elt qui accompagne z dans X parmi les $n-1$ restants.

- Le # de façons d'avoir X avec $z \notin X$ est $\binom{n-1}{2}$ car on doit choisir 2 elt autres que z (donc $n-1$ disponibles).



- $\binom{n}{2}$ est le nb de $X \subseteq S$ avec $|X|=2$ tel que soit $z \in X$, ou $z \notin X$

$$\Rightarrow \binom{n}{2} = \binom{n-1}{1} + \binom{n-1}{2}$$

Preuve par contre-exemple

Prouver un énoncé du type

$$\neg(\forall x, p(x))$$

$p(x)$ = prédicat concernant x

Equivalent à $\exists x, \neg p(x)$

$$\neg(\exists x, p(x))$$

$$\Rightarrow \forall x, \neg p(x)$$

Théorème: le tri sans-gradué présenté en classe ne trie pas toujours correctement.

$\neg (\forall T, \text{tri sans-gradué retourne } T \text{ trié})$
 $\rightarrow (\exists T, \text{tri sans-gradué ne trie pas } T)$

Preuve: on a vu que sur entrée $T = [2 \ 3 \ 1 \ 5 \ 4]$, l'algo retournerait un tableau avec 3 à sa dernière position. Donc, le tableau retourné ne sera pas trié. ■

Preuves par induction

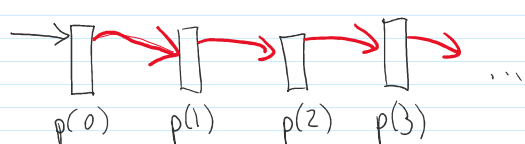
On veut montrer que qq chose est vrai $\forall n \in \mathbb{N}$
(ou parfois pour $x \in \mathbb{N}$)

À montrer: $\forall n \in \mathbb{N}, p(n)$

- On montre que $p(0)$ est vrai.
- Sachant que $p(0)$, on montre que $p(1)$ est vrai. $p(0) \Rightarrow p(1)$
- Sachant que $p(1)$, on montre que $p(2)$ est vrai. $p(1) \Rightarrow p(2)$

...

Plus rapide: • prouver $p(0)$ Cas de base
• prouver $p(n-1) \Rightarrow p(n)$ Induction



↳ Prouver

$p(n) \Rightarrow p(n+1)$ valide

$p(0)$: cas de base

$p(n-1)$: hypothèse d'induction

Théorème: $\forall n \in \mathbb{N}, \sum_{i=0}^n 2^i = 2^{n+1} - 1$ $\sum_{i=0}^n 2^i = 2^0 + 2^1 + 2^2 + \dots + 2^n$

Preuve: (1) Cas de base: $n=0$

$$\text{On a } \sum_{i=0}^n 2^i = \sum_{i=0}^0 2^i = 2^0 = 1$$

$$\text{On a } \sum_{i=0}^1 2^i = \sum_{i=0}^1 2^i = 2^0 = 1$$

$$\text{De l'autre côté, } 2^{n+1} - 1 = 2^{0+1} - 1 = 2 - 1 = 1. \quad \checkmark$$

(2) Induction: $p(n-1) \Rightarrow p(n)$

On suppose par hypothèse d'induction ($p(n-1)$)

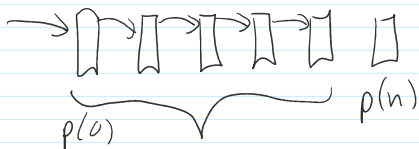
$$\text{que } \sum_{i=0}^{n-1} 2^i = 2^{(n-1)+1} - 1 = 2^n - 1$$

$$\text{On veut prouver que } \sum_{i=0}^n 2^i = 2^{n+1} - 1.$$

$$\begin{aligned} \text{On a } \sum_{i=0}^n 2^i &= \sum_{i=0}^{n-1} 2^i + 2^n & \sum_{i=0}^n 2^i &= 2^0 + 2^1 + \dots + 2^{n-1} + 2^n \\ &= 2^n - 1 + 2^n & & \underbrace{\sum_{i=0}^{n-1} 2^i}_{2^n - 1} + 2^n \quad (\text{par hypothèse}) \\ &= 2 \cdot 2^n - 1 \\ &= 2^{n+1} - 1 \quad \blacksquare \end{aligned}$$

Induction forte

Quand on veut prouver $\forall n \in \mathbb{N}, p(n)$, on peut démontrer que $(p(0) \wedge p(1) \wedge p(2) \wedge \dots \wedge p(n-1)) \Rightarrow p(n)$



Théorème: $\forall n \in \mathbb{N}, n > 0, \exists k, l \in \mathbb{N}$ tels que $n = 2^k \cdot l$
où l est impair.

Preuve: On prouve par induction forte.

(1) Cas de base: $n = 1$. On prend $k = 0, l = 1$
et on a $n = 2^0 \cdot 1 = 1 \cdot 1 = 1$.

(2) Induction. On suppose que $\forall m$ avec $1 \leq m < n$,
l'énoncé est vrai, c'est-à-dire que

(2) Induction. On suppose que $\forall m$ avec $1 \leq m < n$,
l'énoncé est vrai, c'est-à-dire que
 $\exists k_m, l_m$ tel que $m = 2^{k_m} \cdot l_m$. (l_m impair)
On veut prouver que $\exists k, l$ tels que $n = 2^k \cdot l$,
avec l impair.

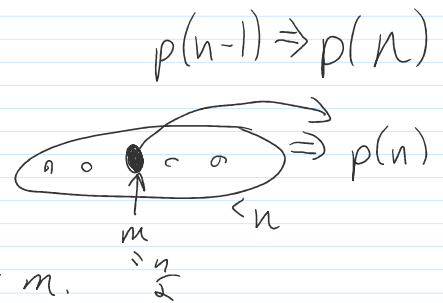
On a 2 cas:

- n est impair: donc $n = 2^0 \cdot n = 1 \cdot n$
(ici, $k=0, l=n$).

- n est pair: donc $n = 2 \cdot m$

On a que $1 \leq m < n$, et on
peut appliquer l'hypothèse sur m .

Donc, $\exists k_m, l_m$ tels que $m = 2^{k_m} \cdot l_m$
 l_m est impair. Donc $n = 2 \cdot 2^{k_m} \cdot l_m$
 $= 2^{k_m+1} \cdot l_m$.



trifusion(T)

si $|T| \leq 1$
return T

A = trifusion(T[1..n/2])

B = trifusion(T[n/2+1..n])

S = []

tant que $|A| \geq 1$ ou $|B| \geq 1$

si $|B| = 0$ ou ($|A| \geq 1$ et $A[1] \leq B[1]$)

| S.append(A[1])

| A.erase(pos=1)

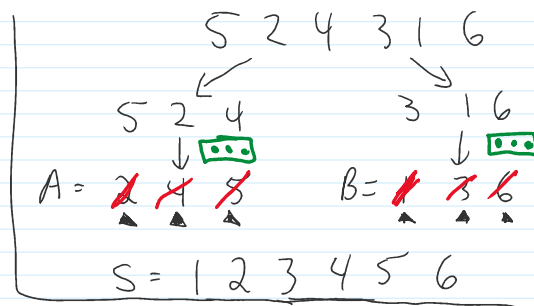
sinon

| S.append(B[1])

| B.erase(pos=1)

x

return S



Lemme: soit T un tableau avec $n \geq 2$ elt's,

Si les tableaux A et B utilisés par le trifusion
sont triés, Alors le S retourné est trié

Si les tableaux A et B utilisés par le tri fusion sont triés, Alors le S retourné est trié et contient chaque elt de A et B.

Exercice

Théorème: le tri fusion trie correctement.

Preuve: par induction sur $n = |T|$.

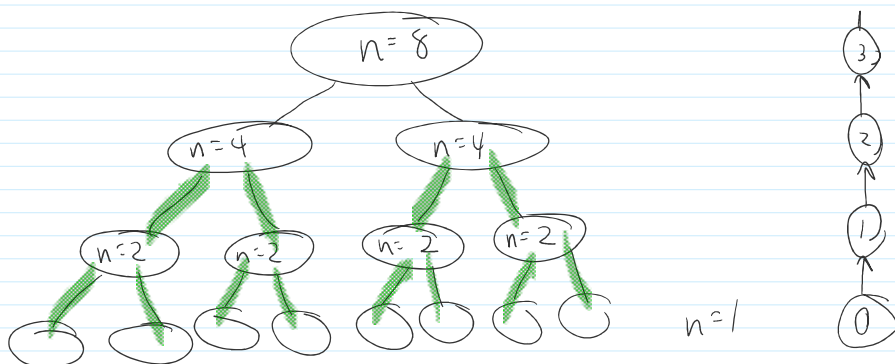
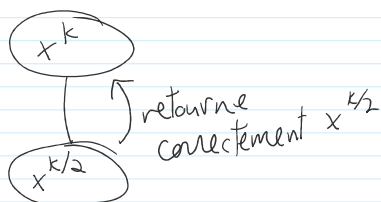
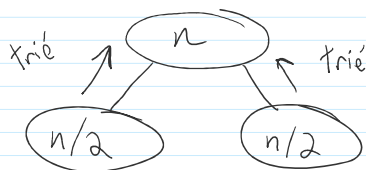
① Cas de base: on fait 2 cas de base.

$n=0$ ou $n=1$, l'algo retourne T, qui sera trivialement trié.

② Induction: on suppose que l'algo trie correctement tout T de taille $< n$.

Par induction, on peut supposer par hypothèse d'induct. les sous-tab. A et B seront triés avant d'entrer dans la boucle de fusion.

Par le lemme, le tableau S retourné sera trié. ▢



Cambien de comparaisons cet algo fait-il, au pire?

$$ACI] \leq BEI]$$

Soit $c(n)$ le # de compar. faites sur entrée de taille n .

$$\text{Si } n=0, c(n)=c(0)=0$$

$$\text{Si } n=1, c(n)=c(1)=0$$

$$\text{Si } n>1, c(n)=c(n/2)+c(n/2)+(\text{\# comp. dans la boucle de fusion})$$



$$= c(n/2)+c(n/2)+n$$

Le # de comp. par tri-fusion est

$$c(n)=\begin{cases} 0 & \text{si } n=0,1 \\ 2c(n/2)+n & \text{si } n>1 \end{cases}$$