TD2 : Algorithmique :

Exercice 1 :

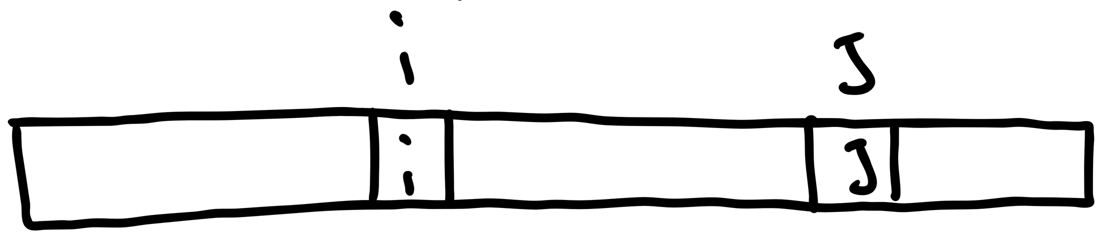
algo naïf :

Pour $i = 0$ à $len(T) - 1$
  [Si $T[i] == i$ Alors retourner i.

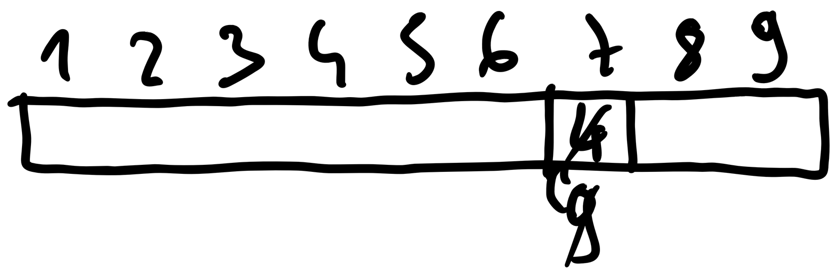
  Retourner $\emptyset$

Complexité $O(n)$
  $n = len(T)$

Propriété remarquable :



-> Si il y a des pts fixes, ils sont sur des "cases" adjacentes

exemple:


$v = 7$ => bingo

$v < 7$ -> rechercher dans 8 --- 14

Sinon $(v > 7)$ -> rechercher dans 0 --- 6

def algorec (T, g, d) :
  Si $g > d$ Alors return -1
  $m = (g+d)/2$
  Si $T[m] = m$ Alors return m
  Sinon si $T[m] < m$ return algorec (T, m+1, d)
  sinon return algorec (T, g, m-1)

Complexités

Master Théorème

$\mathcal{C}(n) = \mathcal{C}(\frac{n}{2}) + 2 \quad n > 0 \quad \rightarrow a = 1 \quad \log_b a = \log_2 1 = 0$

$\mathcal{C}(0) = O(1) \qquad b = 2$

Cas 2 $\rightarrow \mathcal{C}(n) = \Theta(\log n)$

<u>Algo 1</u>(T) :

g = 0

d = len(T) - 1

while (g ≤ d)

m = (g + d) / 2

si T[m] = m , return m

sinon

si T[m] < m :

g = m + 1

sinon

d = m - 1

return -1

Complexité $\Theta(\log n)$

Exercice 2 :

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5$

$x_1 \overset{?}{<} x_2 \qquad$ n élem

n-1 comparaison au minimum

3 .

4. $n = 2^k$

$$C(n) = \begin{cases} 2\,C(\frac{n}{2}) + 2 & n \geq 2 \\ 1 & n = 2 \\ 0 & n = 1 \end{cases}$$

d-g

coût pour $n = 2^k$

$$d_k = \begin{cases} 2\,d_{k-1} + 2 & \text{si } k \geq 2 \\ 1 & \text{si } k = 1 \\ 0 & \text{si } k = 0 \end{cases}$$

$d_k = ?$

$$d_k = 2\,d_{k-1} + 2$$

$$= 2(2d_{k-2} + 2) + 2$$

$$= 2(2(2d_{k-3} + 2) + 2) + 2$$

$$= 2^{k-1} \cdot \underbrace{d_1}_{=1} + \sum_{i=1}^{k-1} 2^i \rightsquigarrow \frac{1 - 2^{k-1}}{1 - 2} \times 2$$

coût pour $n = 2^k$

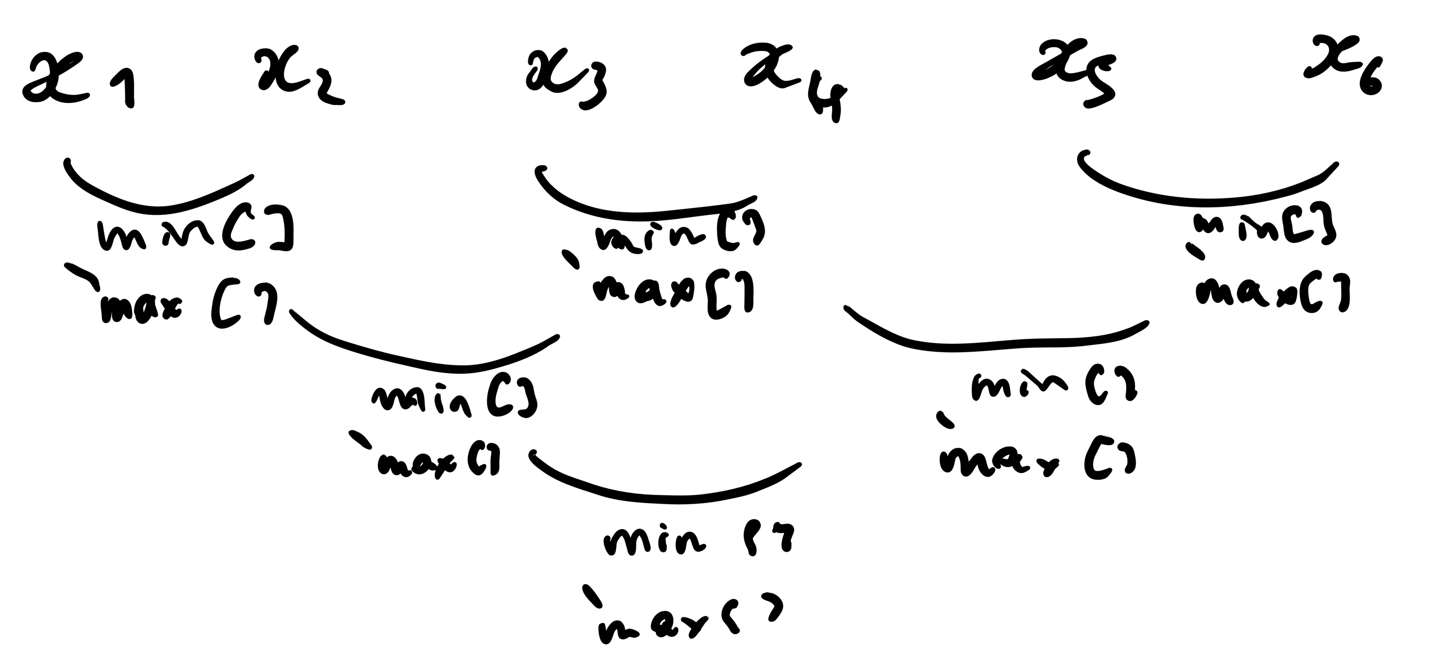
$$d_k = 2^{k-1} + 2^k - 2$$

$$= \frac{3}{2} 2^k - 2$$

$$\boxed{C(n) = \frac{3}{2} n - 2}$$

Bis : Montrer que c'est optimal

5.



$\frac{n}{2}$

$\frac{n}{2}-1 \rightarrow$ min

$\frac{n}{2}-1 \rightarrow$ max

ajoute les élements dans une liste du min / max

- au debut = $P_{\{1,..,n\},\{1-n\}}$ = tout le monde est candidat à tout
- l'obj : arriver à $P_{\{i\},\{j\}}$ = i est le min
  j est le max

optimal car au bout d'un moment obligé de séparer en deux paquet

$\Theta(n)$ //Master théorème

Exo 3:

1) $A(n) = A\left(\frac{9n}{10}\right) + 1$

$\log_8 9 = \frac{\ln 9}{\ln 8}$

$a = 1$

$b = \frac{10}{9}$

$\log_b a = \log_{\frac{10}{9}}(1) = 0$

$f(n) = n$ $\quad \in \Omega(n^{\varepsilon})$ pour $0 < \varepsilon < 1$

$c < 1$ tq. $a.f\left(\frac{n}{b}\right) \leq c.f(n)$?

$\frac{9n}{10} \leq c.n.$ oui

2) $a = 2$

$b = 4$

$\log_4 2 = \frac{1}{2}$

$f(n) = \sqrt{n} = n^{\frac{1}{2}}$

CAS 2: $B(n) = \Theta(\sqrt{n}.\log(n))$

//Finir C, D, E, F

↳ Exo 4