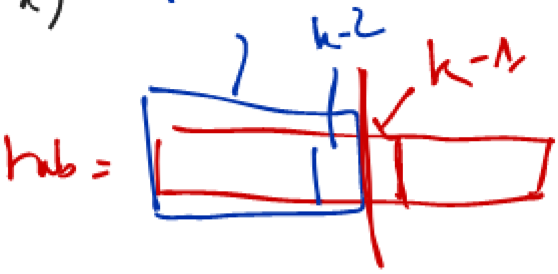
TD 3

Exercice 1

```
def nb_occ(tab, k, x):
    print('tab=', tab[0:k], 'x=', x)
    if k == 0:
        res = 0
    else:
        if tab[k-1] == x:
            res = nb_occ(tab, k-1, x)+1
        else:
            res = nb_occ(tab, k-1, x)
    print('tab=', tab[0:k], 'x=', x, 'nboccurrences=', res)
    return res
```

nb_occ(tab, k-1, x)



Q1 tab = [3, 6, 7, 6, 2, 6, 3], k = 7, x = 6.

"+1" à chaque fois qu'on voit 6

tab = [3,6,7,6,2,6,3] x = 6
tab = [3,6,7,6,2,6] x = 6
tab = [3,6,7,6,2] x = 6
tab = [3,6,7,6] x = 6
tab = [3,6,7] x = 6
tab = [3,6] x = 6
tab = [3] x = 6
tab = [] x = 6
tab = [] x = 6 nboccurrence = 0
tab = [3] ——— = 0
tab = [3,6] ——— 1
tab = [3,6,7] ——— 1
tab = [3,6,7,6] ——— 2

nb_occ(tab, 7, 6)
↳ res = nb_occ(tab, 6, 6)
nb_occ(tab, 5, 6) + 1
nb_occ(tab, 4, 6) (2)
nb_occ(tab, 3, 6) + 1 (2)
nb_occ(tab, 2, 6) (1)
nb_occ(tab, 1, 6) + 1 (1)
nb_occ(tab, 0, 6) = 0

tab = [3,6,7,6,2] x = 6 nb.occ. = 2
tab = [3,6,7,6,2,6] ——— 3
$\boxed{\text{tab} = [3,6,7,6,2,6,3] ——— 3}$

$\boxed{Q2}$ $P(k)$: nb_occ(tab, k, x) termine et renvoie le nombre d'occurrences de x entre les indices 0 et k-1 inclus.

. Base : $P(0)$

nb_occ(tab, 0, x) termine car ne fait pas d'appel récursif ; et renvoie 0.

$P(1)$ nb_occ(tab, 1, x) appelle

(nb_occ(tab, 0, x) et vaut 0 si tab[0] ≠ x et vaut 1 si tab[0] == x.
et termine car nb_occ(tab, 0, x) termine).

. Induction : Soit $k \geq 1$. on suppose $P(k-1)$

nb_occ(tab, k, x) fait appelle à nb_occ(tab, k-1, x) donc termine par HR.

+ facile pour se ramener au code de la fonction

2 cas

. si tab[k-1] == x, alors nb_occ(tab, k, x) renvoie nb_occ(tab, k-1, x) + 1

Or par HR nb_occ(tab, k-1, x) renvoie le nombre d'occurrences de x dans tab[0,..., k-2] et tab[k-1] == x donc nb_occ(tab, k-1, x) + 1 est le nombre d'occurrences de x dans le tab[0.., k-1].

. Si tab[k-1] ≠ x, nb_occ(tab, k, x) renvoie nb_occ(tab, k-1, x) qui est le nombre d'occurrences dans tab[0,..., k-2] et comme tab[k-1] ≠ x, c'est le nombre d'occurrences de x dans le tableau tab $[0, \dots, k-1]$

Conclusion : $P(k)$ est initialisée et héréditaire donc vrai pour tout $k \in \{\underset{1}{\cancel{0}}, \dots, n\}$

Exercice 4

$$m_n(x) = \begin{cases} 1 & \text{si } n = 0 \\ \left(m_{\frac{n}{2}}(x)\right)^2 & \text{si } n \text{ est pair} \\ \left(m_{\frac{n-1}{2}}(x)\right)^2 \times x & \text{si } n \text{ est impair} \end{cases}$$

Q1

1) $$m_{\underset{\text{pair}}{10}}(2) = \left(m_{\underset{\text{impair}}{5}}(2)\right)^2 = \left(\left(m_2(2)\right)^2 \times 2\right)^2$$

$$= \left(m_{\underset{\text{pair}}{2}}(2)\right)^4 \times 2^2$$

$$= \left(m_1(2)^2\right)^4 \times 2^2$$

$$= \left(m_{\underset{\text{impair}}{1}}(2)\right)^8 \times 2^2$$

$$= \left(\left(\underbrace{m_0(2)}_{=1}\right)^2 \times 2\right)^8 \times 2^2$$

$$= 2^8 \times 2^2$$

Donc $\boxed{m_{10}(2) = 2^{10}}$

2) · Base $u_0(x) = 1 = x^0$ ✓

· Induction

Soit $n \geq 1$, on suppose que $\forall k \leq n-1 \quad u_k(x) = x^k$

Alors

→ si n est pair, $n = 2k$, $u_n(x) = \left(u_{\frac{n}{2}}(x)\right)^2$, $\frac{n}{2} \leq n-1$

donc $u_{\frac{n}{2}}(x) = x^{\frac{n}{2}}$ donc $u_n(x) = \left(x^{n/2}\right)^2 = x^n$ ✓

→ si n est impair, $u_n(x) = \left(u_{\frac{n-1}{2}}(x)\right)^2 \times x$,

et $\frac{n-1}{2} \leq n-1$ donc $u_{\frac{n-1}{2}}(x) = x^{\frac{n-1}{2}}$

donc $u_n(x) = \left(x^{\frac{n-1}{2}}\right)^2 \times x = x^{n-1} \times x = x^n$ ✓

· Conclusion : La propriété est initialisée

et $\forall n \geq 1 \left(\left(\forall k < n \quad u_k(x) = x^k \right) \Rightarrow u_n(x) = x^n \right)$

Donc, par récurrence forte, $\forall n \geq 0 \quad u_n(x) = x^n$.

Made with Ziteboard