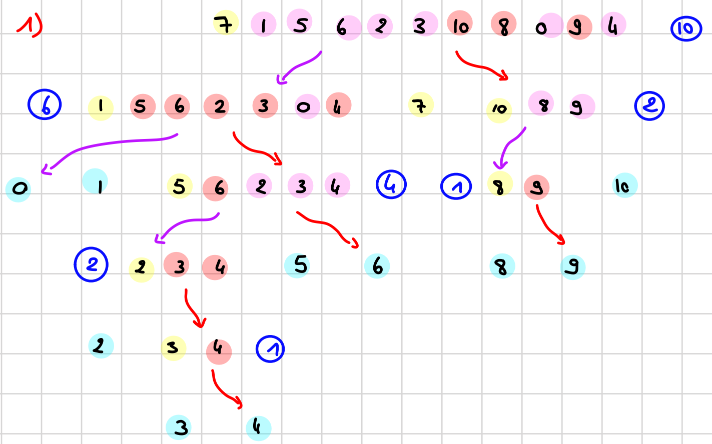
15.03.2021 (présentiel)

EA4 TD6

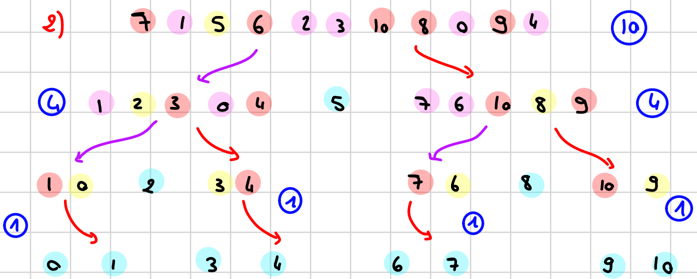
Exercice 1

1)



$\Rightarrow$ 26 comparaisons

2)



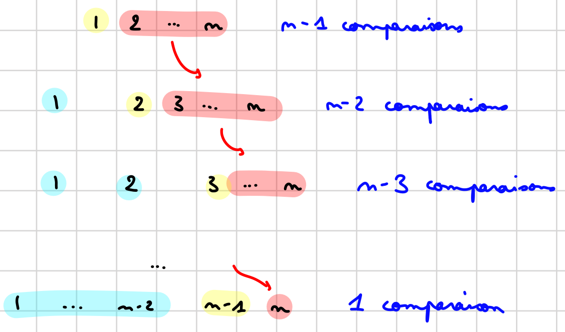
$\Rightarrow$ 22 comparaisons

trouver la médianne se fait en $\Theta(n)$

$$c(n) = \underbrace{2\,c\left(\frac{n}{2}\right)}_{\text{tri des sous-tableaux}} + \overbrace{\Theta(n)}^{\text{partition}} = \Theta(n\log(n))$$

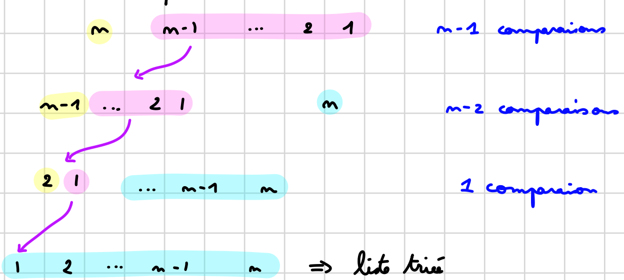
Exercice 2

1) $C[n] = [1, 2, \dots, n-1, n] \Rightarrow \sum_{i=0}^{n-1} i = \frac{(n-2)(n-1)}{2} = \frac{1}{2}(n^2 - 3n + 2) = O(n^2)$ comparaisons.



$\Rightarrow$ liste triée

$D[n]$: même chose que $C[n]$



$\Pi[k] = [2**k] + \Pi[k-1] + [i + 2**k \text{ for } i \text{ in } \Pi[k-1]]$ avec $\Pi[0] = 1$

$\boxed{2**k = 2^k}$

ex. $\Pi[0] = [1]$ $\Pi[2] = [4, 2, 1, 3, 6, 5, 7]$

$\Pi[1] = [2, 1, 3]$ $\Pi[3] = [8, 4, 2, 1, 3, 6, 5, 7, 12, 10, 9, 11, 14, 13, 15]$

3) Hauteur max : n → si le tableau est déjà trié

Hauteur min : $\log(n)$ → si on tombe sur l'élément médian pour chaque pivot.

4)

meilleur cas :

Pour un étage e donné, $c(n) = 2^e \times \left(\frac{n}{2^e} - 1\right) = n - 2^e$

Donc pour une profondeur p : $c(n) = n - 2^p + n - 2^{p-1} + \dots + n - 1$

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

Exercice 4

1)
```
def tri-stable (T):
    bleu = []
    rouge = []
    blanc = []
    for x in T:
        if estBleu (x):
            bleu += [x]
        else if estBlanc (x):
            blanc += [x]
        else if estRouge (x):
            rouge += [x
    return bleu + blanc + rouge
```

2)
```
def tri-linéaire (T):
    i = 0
    j = T.length -1
    while i <= j :
        if estBlanc (T[i]):
            i++
        else :
            T[i], T[j] = T[j], T[i]
            j--
```

3) l'algo fonctionne s'il n'y a qu'un seul élément de couleur blanche.

4)
```
def tri-3-couleurs (T):
    i = 0
    j = 0
    k = T.length -1
    while j <= k :
        if isBleu (T[j]):
            échanger (T[i], T[j])
            i ++
            j ++
        if isBlanc (T[j]):
            j ++
        if isRouge (T[j]):
            échanger (T[j], T[k])
            k --
```

BLEU	BLANC	NON-EXPLORÉ	ROUGE

0 i j k T.length