Complexité

- complexité temporelle

(• RAM prise)

```
for i in range(n):
    j = 0
    while j*j < i:
        x += 1
        j += 1
```

$\sim\sqrt{n}$; $\times n$ $\Rightarrow O(n\sqrt{n})$

```
i, x = n, 0
while i > 1:
    i += 1
    i //= 2
```

arrêt qd $\frac{n}{2^{\text{nb. d'its}}} = 1 \Leftrightarrow 2^{\text{nb. it}} = n$

$\Leftrightarrow$ nb it $= \log_2 n$ $\Rightarrow O(\log n)$

Hot tip! $O(\log n)$ c'est souvent qd on divise la var d'it. à chaque it.

```
x, i = 0, n
while i > 1:
    for j in range(i):
        x += 1
    i //= 2
```

$O(n)$

dichotomie: $O(\log n)$

Au pire des cas, je coupe nb_{it} fois la liste en 2 pour avoir liste de taille 1

$$\frac{n}{2^{nb_{it}}} = 1 \quad \text{pour l'arrêt}$$

ie $O(\log n)$

Valide comme explication/rédaction pour un DS

$C(n)$: complexité moyenne pas à réviser