

# Complexité : exemple des tris...sans les tris.

Académie de Lyon

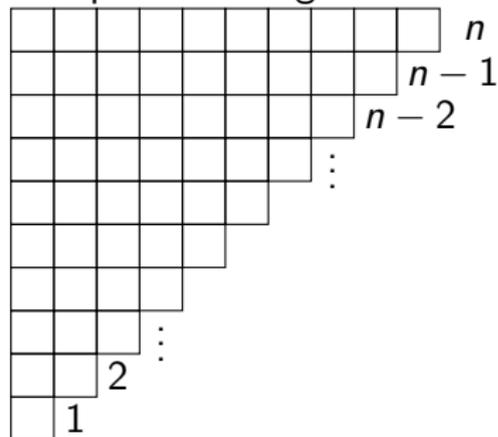
# Un algorithme qui ne fait rien... mais qui le fait en temps quadratique

Que dire du temps d'exécution de :

```
def triangle(n) :  
    for i in range(n) :  
        for j in range(i,n) :  
            pass
```

# Le triangle.

Complexité triangulaire :



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

Temps d'exécution proportionnel à  $\frac{1}{2}n(n+1)$ .

Contrôle expérimental

# Un algo un peu moins inutile, svp

**Exercice.** Écrire un algorithme de tri d'une liste dont la complexité relève du schéma triangulaire précédent.

Exemple du tri par sélection (algorithme au programme de la spécialité ISN)

# Un algo un peu moins inutile, svp

Principe du tri par sélection d'une liste  $L[1], L[2], \dots, L[n]$  :

Pour  $j$  de 1 à  $n-1$  :

repérer l'élément minimum dans  $L[j+1], L[j+2], \dots, L[n]$

échanger  $L[j]$  avec cet élément minimum

Tri par sélection

# Un algorithme qui ne fait rien... mais plus vite !

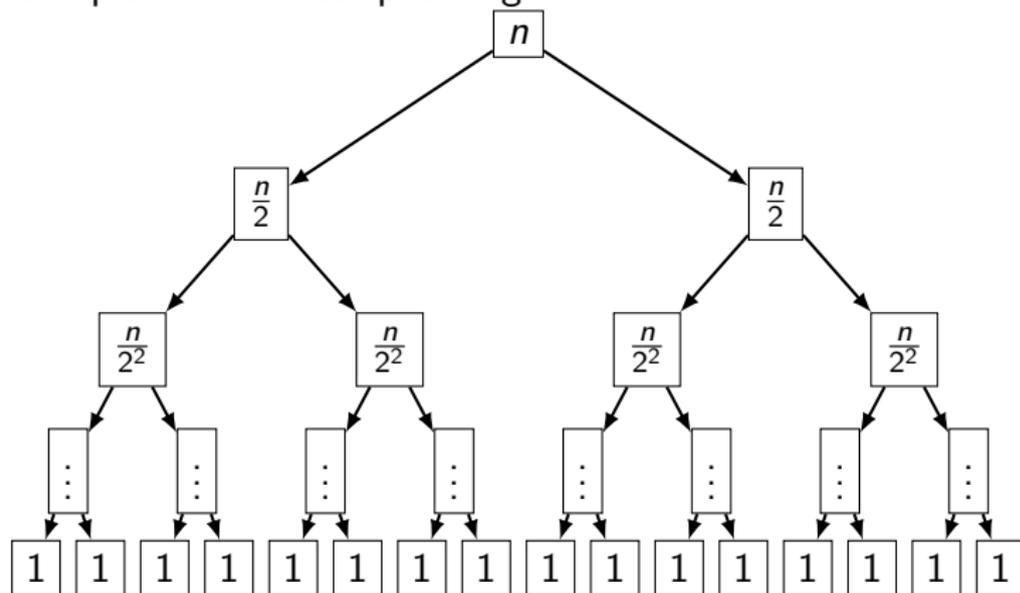
Que dire du temps d'exécution de :

```
def arbre(n) :  
    h, t=1, n  
    while t >= 1 :  
        for j in range(h):  
            for k in range(t):  
                pass  
  
        h=2*h  
        t=t/2
```

contrôle expérimental

# Un algorithme qui ne fait rien... mais plus vite !

Complexité « diviser pour régner ».



# Arbre.

$$1 \times n + 2 \times \frac{n}{2} + 2^2 \times \frac{n}{2^2} + \cdots + 2^k \times \frac{n}{2^k} = k \times n \text{ avec } 2^k = n$$

$$1 \times n + 2 \times \frac{n}{2} + 2^2 \times \frac{n}{2^2} + \cdots + 2^k \times \frac{n}{2^k} = \log_2(n) \times n$$

Temps d'exécution proportionnel à  $n \log_2(n)$ .

contrôle expérimental

# Un algo un peu moins inutile, svp

**Exercice.** Écrire un algorithme de tri d'une liste dont la complexité relève du schéma précédent.

Exemple du tri par fusion (algorithme au programme de la spécialité ISN).

# Un algo un peu moins inutile, svp

## Principe du tri par fusion.

Si la liste contient au plus un élément, elle est triée. Sinon :

- 1 On la partage en deux listes gauche et droite de longueurs à peu près égales.
- 2 On trie la sous-liste gauche et la sous-liste droite ainsi obtenues.
- 3 On fusionne ces deux sous-listes.

une traduction python