Seminaire Complexité 1

$\Theta \Rightarrow$ Borne supérieure

$(3n+7)\times(4n+7n-5) = \Theta(n) \times \Theta(n^2) = \Theta(n^3)$

$T(n) = an^2 + bn + c$

$O(n)$ = une fonction qui est | au pire linéaire
| donnée par une fonction linéaire.

$3n+2 = O(n)$ $\quad 4 = O(n)$ $\quad$ O(n) est une fonction linéaire
On ne precise pas laquelle

$3n+2 = O(n^2)$

$O(n) + O(n) = O(n)$: Une fonction linéaire + une f° lin => une f° linéaire.

$O(n) + O(n^2) = O(n^2)$: Une f° linéaire + Une f° quadratique : Au pire quad.

$O(n) \times O(n^2) = O(n^3)$: f° linéaire x f° quad = f° cubique.

$\Theta(n+n^2) = \Theta(n^2) :=: \Theta(f(n)+g(n)) = \Theta(\max(f(n); g(n)))$

$O(n+n^2) = O(n^2)$!!

$\Omega(n)$: une fonction qui domine une f° linéaire.

$n^2 = \Omega(n)$ $\quad$ n^2 domine n'importe quelle f° linéaire.

$O(n) \times \Theta(n^2) = O(n^3)$

$2n \times (4n^2 + 2n) = 8n^3 + 4n^2$

$\sqrt{n} \times n^2 = n^{2,5}$

$\Theta(n) - \Theta(n) = O(n)$

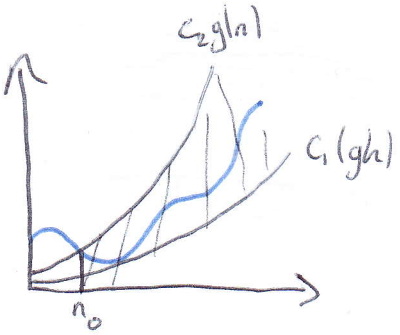
$2n - n = n$

$2n - (2n-1) = 1$

$2n - 2n = 0$

$n - 2n = -n$

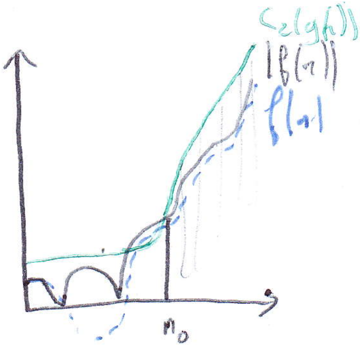
<u>Definition:</u> $\Theta(g(n)) = \{ f(n) \mid \exists c_1 > 0\ \exists c_2 > 0, \exists n_0 \in \mathbb{N},$
$\forall n \geq 0,\ c_1 g(n) \leq f(n) \leq c_2 g(n) \}$



$3n^2+n \in \Theta(n^2)$: $\underbrace{c_1}_{3} n^2 \leq 3n^2 + n \leq \underbrace{c_2}_{4} n$

<u>$O(g(n))$</u> : $\{ f(n) \mid \exists c_2 > 0, \exists n_0 \in \mathbb{N}, \forall n \geq 0, |f(n)| \leq c_2 g(n) \}$

$10\cos(n) = O(1) \Rightarrow \underset{n_0=0}{|10\cos(n)|} \leq \underbrace{c_2}_{10} \times 1$



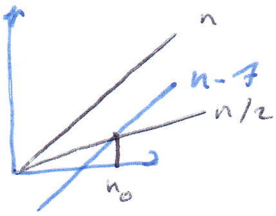
Scanné par Hyperion

$f(n) = \Theta(g(n))$ au lieu de $f(n) \in \Theta(g(n))$ par abus de notation.

De même $\Theta(n) + \Theta(n^2) = \Theta(n^2) \Rightarrow$ Ici $\Theta(n) + \Theta(n^2) \in \Theta(n^2)$.

SelectionSort(A, n):

for i ← 0 to n-2	$\Theta(n)$	$n - 7 = \Theta(n)$
min pos ← i	$\Theta(n)$	$c_1 n \leq n - 7 \leq c_2 n$
for j ← i+1 to n-1	$\Theta(n^2)$	$c_1 = \frac{1}{2}$, $c_2 = 1$
if A[j] < A[min pos]:	$\Theta(n^2)$	$\sum_{i=0}^{n-2} \underbrace{\sum_{j=i+1}^{n-1}}_{n-i-1}$
min pos ← j	$\Theta(n^2)$	
A[i] ↔ A[min pos]	$\Theta(n)$	
endfor		
end.		



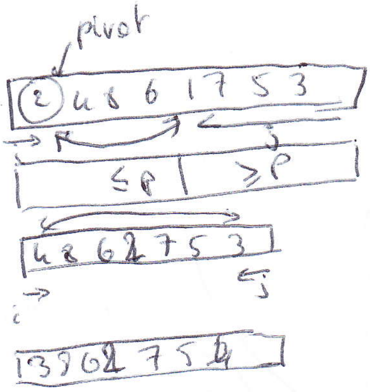
Insertion Sort:

Best	Moy	Worst
$\Theta(n)$	$\Theta(n^2)$	$\Theta(n^2)$

En général, on peut dire $T(n) = O(n^2)$

cette notation ne garantie pas qu'un cas est quadratique.

HeapSort(A, n):
```
  BuildHeap(A, n)
  for i ← n-1 to 1:
     A[0] ↔ A[i]
     Heapify(A, 0, i)
```



QuickSort(A, b, e):
```
// Trie A[b..e-1].
if e - b > 1
   m ← Partition(A, b, e)
   // [b     m     e]
   QuickSort(A, b, m)
   QuickSort(A, m, e)
```

Partition(A, b, e):
```
   i ← b - 1 ; p → A[b].
   j ← e
   for ever:
      ++i
      while A[i] < p:
         ++i
      --j
      while A[j] > p:
         --j
      if i < j
         A[i] ↔ A[j].
      else
         return i + (b==i)
```

Scanné par Hyperion