CRESCITA DELLE FUNZIONI (NOTAZIONI)

$T: \mathbb{D} \to \mathbb{C}$ — $d \in \mathbb{D}$ ⇝ DIMENSIONE DEGLI INPUT

$T(d)$, con $d \in D$ ⇝ COSTO COMPUTAZIONALE

$T(n)$ ⇝ costo tempo $T(s)$ ⇝ costo spazio

ESEMPI:

$f(n) = 3n^2$ $f(n) \simeq \Theta(n^2)$ (n^2 → $g(n)$)

perché $\exists c_1, c_2 > 0$ $n_0 \in \mathbb{N}$ t.c. $\underbrace{c_1}_{3-k} g(n) \leq f(n) \leq \underbrace{c_2}_{3+k} g(n)$

↳ perché: $(3-k)\, g(n) \leq 3n^2 \leq (3+k)\, g(n)$
SEGUE LA NOTAZIONE

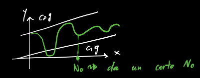
ESSENDO $\Theta(n^2)$ → È ANCHE $O(n^2)$ e $\Omega(n^2)$ { DEFINIZIONI "COMPRESE"

COMPORTAMENTO ASINTOTICO DI $T(n)$ ed $S(n)$:

Θ, O, Ω Sia g una $g: \mathbb{N} \to \mathbb{N}$, quindi

CON "ABUSO DI NOTAZIONE", POSSO SCRIVERE ⇝ $f(n) = \Theta(g(n))$ oppure $f(n) = O(g(n))$ oppure $f(n) = \Omega(g(n))$

(1) $\Theta(g(n))$ → LIMITE ASINTOTICO STRETTO → RAPPRESENTA: (L'INSIEME DELLE FUNZIONI)
↳ $\{ f \mid \exists c_1, c_2 > 0 \ \exists n_0 \in \mathbb{N} \mid \forall n \geq n_0$ t.c. $\underline{c_1 g(n) \leq f(n) \leq c_2 g(n)} \}$



(2) $O(g(n))$: $\{ f \mid \exists c_0 \ \exists n_0$ t.c. $\forall n \geq n_0$ $f(n) \leq c_0 g(n) \}$ → on the Book :D

$f \in O(g(n))$

(3) $\Omega(g(n)) = \{ f, \exists c > 0 \ \exists n_0 \in \mathbb{N}$ t.c. $\forall n \geq n_0$ $f(n) \geq c \cdot g(n) \}$

2)
$f(n) = 4n^2 - 2n$
$= \Theta(n^2)$

$c_1 n^2 \leq 4n^2 - 2n \leq c_2 n^2$

$\exists c_1, c_2 > 0$ $\exists n_0 \in \mathbb{N}$ t.c. $c_1(n^2) \leq f(n) \leq c_2(n^2)$

$f(n) = 4n^2 - 2n \leq 4n^2$ $c_2 = 4$

$4n^2 - 2n \geq 2n^2$

REGOLINA: $f(n) = P_k(n)$, con n^k grado massimo, ⇝ $\Theta(n^k)$

$f(n) = k = \Theta(1)$

Altre f $\Theta(1)$, esempio: $\{ \ldots$ costanti $\ldots \}$ e quello che riesce ad essere costante / periodico che ⇝ $\Theta(1)$

$14 \cdot 1 \leq \sin(n^k) + 16 \leq 15$

$(-1)^n + 16$

ESERCIZIETTO (sul libro), trova c_1 e c_2 della notazione.

$f(n) = 4n^2 - n\log(n) = \Theta(n^2)$

$c_1(4n^2 - n\log(n)) \leq 4n^2 \leq c_2(4n^2 - n\log(n))$

$c_2(4n^2 - n\log(n)) \leq 4n^2$; $c_2 = 4$

↳ $\log(n) < n$
$n\log(n) < n^2$
$-n\log(n) \geq -n^2$

$4n^2 - n\log(n) > -n^2$

$f(n) = \log(n) = \Theta(\log(n))$
$= O(n)$
$= \Omega(1)$
→ limiti della funzione logaritmica.

Ciao sono Daniel

$f(n) = 6n\log(n) = \Theta(n \log n)$
$O(n^2)$
$\Omega(n)$

Hello

RIPASSO VALUTAZIONE TEMPI DI CALCOLO DELLE PR. IT:

$\boxed{T(n)}$ n = dimensione input   Assunzione: "ogni istruzione ha lo stesso peso"

$[a; b] = b - a + 1$

```
x := 1                 → 1
for i = 2 to n         → cf1 = (n-2+1) + istruzione (n+1) = n-2+1+1 = n
    for J = 1 to i     → cf2 = i+1 = Σ_{i=2}^{n} i + Σ_{i=2}^{n} 1 = n(n+1)/2 - 1 + (n-2+1)·1
        x = x + i + J  → c2 = cf2 - 1
return x               → 1
```

istruzione $(n+1)$: stop loop → vale 1

-1: perché parto da 2

$c_2 = cf_2 - 1$: perché non ho istr. stop.

$i = 2 \to J \in [1, 2] = 2 - 1 + 1 + 1 = 3$
$i = 3 \to J \in [1, 3] = 3 - 1 + 1 + 1 = 4$
$\ldots$
$i \to J \in [1, i] = i - 1 + 1 + 1 = i + 1$

$T(n) = 1 + \underset{cf_1}{n} + \underbrace{\frac{n(n+1)}{2} - 1 + n - 1}_{cf_2} + \underbrace{\frac{n(n+1)}{2} - 1}_{\text{op. interne}}$

$= \Theta(n^2)$

Ripasso ricorsività (esempi)

1) $f: \mathbb{N} \to \mathbb{N}$ (FATTORIALE DI N)

$$f(n) = \begin{cases} 1 & \text{se } n = 0 \\ n \cdot f(n-1) & \text{se } n > 0 \end{cases}$$

$\boxed{\text{FATT}(n)}$ → Algoritmo

```
if n=0
   return 0
else
   return n · fatt(n-1)
```

FIB: → esercizio (fai stare).

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

2) $f: \mathbb{N} \to \mathbb{N}$ → FIBONACCI SUCCESSIONE

$$f(n) = \begin{cases} n & \text{se } n = 0 \vee n = 1 \\ f(n-1) + f(n-2) & \text{se } n > 1 \end{cases}$$

```
Fib:
if (n ≤ 1):
    return n;
else:
    return fib(n-1) + fib(n-2);
```

$f(n) = n^2$ → Riscriverlo ricorsivamente (2° esercizietto da ripassare)

$f(n) = n^2$

$f(n-1) = (n-1)^2 = n^2 - 2n + 1$

$f(n-2) = \underbrace{n^2 - 2n + 1}_{f(n-1)} + 2n - 1$

→ * FORMULA RICORSIVA:

$$\begin{cases} 0 & \text{se } n = 0 \\ f(n) = f(n-1) + 2n - 1 & \text{se } n > 0 \end{cases}$$

Quad.

```
if (n = 0) {
    return 0;
} else {
    return quad(n-1) + 2n - 1;
}
```

* Tempo di calcolo:

$n > 0$

$T(n) = \underbrace{1}_{\text{if/else}} + 1 + T(n-1)$; (return)

$$T(n) = \begin{cases} 2 & \text{se } n = 0 \\ 2 + T(n-1) & \text{se } n > 0 \end{cases}$$

$= 2 + T(n-1)$

$= 2 + 2\ T(n-2)$

$= 2 + 2 + 2 + T(n-3)$

$= \ldots 2 + 2 \ldots + T(1)$

$= 2 + \ldots + 2 + 2 \Rightarrow 2(n) + 2$ (srotolamenti)

* TECNICA DIVIDE ET IMPERA:

1) Dividere in k sottoproblemi
2) Risolvere ricorsivamente i
3) Combinare le soluzioni dei sott. → produco la soluz. al probl. dato.

ESEMPIO: Ordinamento $\vec{V}$ → strategia quicksort / mergesort.

```
mergeSort (A, l, r):
if (l < r)
    m = [(l+r)/2];
    mergesort (A, l, m);
    mergesort (A, m+1, r);
    merge (A, l, m, r);
```

* Tempo di Calcolo:

$n > 1$

$T(n): \underbrace{1}_{\text{Test}} + \underbrace{1}_{\text{Calcolo } m} + \underbrace{T(n/2)}_{1^a \text{ c.r.}} + \underbrace{T(n/2)}_{2^a \text{ c.r.}} + \Theta(n)$

$: 2T(\frac{n}{2}) + \Theta(n)$

Notazione Asintotica, calcolo dei tempi di un algoritmo iterativo e ricorsivo, tecnica divide-et-impera