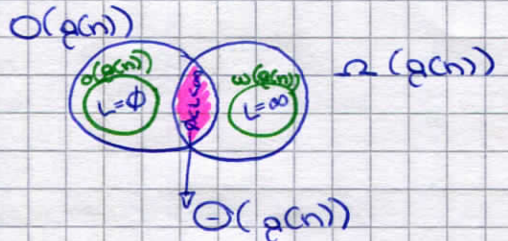
2.10.13

Per la notazione asintotica si possono utilizzare anche i limiti.



CLASSI DI COMPORTAMENTO ASINTOTICO

LIMITE DI UNA SUCCESSIONE

$\{a_n\}_{n \in \mathbb{N}}$ Studiare il comportamento asintotico = calcolare il limite

$\lim_{n \to \infty} a_n = L \in \mathbb{R}$ "Comunque io prenda n, la successione a_n si avvicina sempre di più a L"

DEFINIZIONE DI LIMITE (SE LA SUCCESSIONE CONVERGE)

$\forall \varepsilon > \emptyset \quad \exists n_\emptyset \in \mathbb{N} \ \ni' \ \forall n \geq n_\emptyset : \quad \underbrace{|a_n - L|}_{\text{distanza tra } a_n \text{ e } L} \leq \varepsilon$

$\emptyset \leq \frac{f(n)}{g(n)} < c$

$\emptyset \leq a_n \leq \varepsilon$

$f(n) = o(g(n)) \iff \lim_{n \to \infty} \frac{f(n)}{g(n)} = \emptyset \overset{*}{=} f(n) = \underbrace{O(g(n))}$ ↳ perché o-piccolo è sottoinsieme di O-grande (non vale il contrario)

DEFINIZIONE DI LIMITE (SE LA SUCCESSIONE NON CONVERGE)

$\lim_{n \to \infty} a_n = +\infty$ "Comunque prenda n, da un certo punto in poi la successione sarà più grande"

$\forall c > \emptyset \quad \exists n_\emptyset \in \mathbb{N} \ \ni' \ \forall n \geq n_\emptyset : \quad c \leq a_n$

$\omega(g(n)) = \emptyset \leq c < \frac{f(n)}{g(n)}$

$f(n) = \omega(g(n)) \iff \lim_{n \to \infty} \frac{f(n)}{g(n)} = +\infty = f(n) = \underbrace{\Omega(g(n))}$ ↳ come prima

RIASSUMENDO

$\forall c > \emptyset \ \exists n_\emptyset \in \mathbb{N} \ \ni' \ \forall n \geq n_\emptyset :$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \begin{cases} \emptyset \Rightarrow f(n) = O(g(n)) = o(g(n)) \\ L \ (\text{con } \emptyset < L < +\infty) \Rightarrow f(n) = \Theta(g(n)) \\ \infty \Rightarrow f(n) = \Omega(g(n)) = \omega(g(n)) \end{cases}$$

DIMOSTRO CHE:

$\lim_{n \to \infty} \frac{f(n)}{g(n)} = L \in \mathbb{R} \ [0 < L < +\infty] \Rightarrow f(n) = \Theta(g(n))$ ⊛

$\lim_{n \to \infty} \frac{f(n)}{g(n)} = L \Rightarrow O(g(n))$

Se $L = \emptyset$ sarebbe o-piccolo $\Rightarrow O$

Se $L > \emptyset$ sarebbe nell'intersezione

PER DEFINIZIONE $\left| \frac{f(n)}{g(n)} - L \right| < \varepsilon$

È vero se

$$\varepsilon < \frac{f(n)}{g(n)} - L < \varepsilon$$

$$(L-\varepsilon) < \frac{f(n)}{g(n)} < (L+\varepsilon)$$

$$(L-\varepsilon)\cdot g(n) < f(n) < (L+\varepsilon)\cdot g(n)$$

Se è vero per ogni ε, significa che è vero anche per $\varepsilon = 1$

ottengo $(\underbrace{L-1}_{c1})\cdot g(n) < f(n) < (\underbrace{L+1}_{c2})\cdot g(n)$

mi ricorda c1 e c2!

Quindi ho dimostrato ★

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = L \quad \Rightarrow f(n) = \Omega(g(n))$$

⟶ come sopra

ESERCIZIO

$(n+a)^b = \Theta(n^b)$

$$\lim_{n\to\infty} \frac{(n+a)^b}{(n)^b} = \lim_{n\to\infty} \left(\frac{n+a}{n}\right)^b = \lim_{n\to\infty} \left(\overbrace{1}^{1} + \overbrace{\frac{a}{n}}^{\emptyset}\right)^b = 1 = L$$

ESERCIZIO

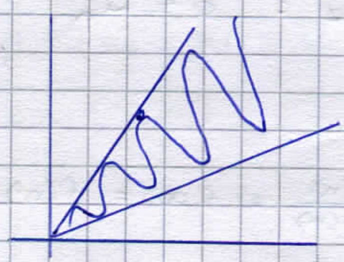
$f(n) = n(\overbrace{1+\sin(n)}^{\leq 2})$

$g(n) = n$

$n(1+\sin(n)) \leq 2n$

$n(1+\sin(n)) = O(n)$

proviamo col limite



$$\lim_{n\to\infty} \frac{\cancel{n}(1+\sin(n))}{\cancel{n}} = \lim_{n\to\infty} 1 + \sin(n)$$ MA $\sin(n)$ non ha limite!

$f(n) = \Theta(g(n))$

ESERCIZIO

$f(n) = \overbrace{n^8}^{\text{prevale}} + 7n^7 - 10n^5 - 2n^4 + 3n^2 - 17 = \Theta(n^8)$ ⟶ Questo algoritmo ha complessità n^8

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{n^8 + 7n^7 - 10n^5 - 2n^4 + 3n^2 - 17}{n^8} = \lim_{n\to\infty}\left(1 + \underbrace{\frac{7}{n} - \frac{10}{n^3} - \frac{2}{n^4} + \frac{3}{n^5} - \frac{17}{n^8}}_{=\emptyset}\right)$$

$\varphi(n)$ è polinomio di ordine k, allora $\varphi(n) = \Theta(n^k)$

ESERCIZIO

$\log(n) \qquad \sqrt{n}$

$$\lim_{n\to\infty} \frac{\log(n)}{\sqrt{n}} \overset{\text{Hôpital}}{=} \lim_{n\to\infty} \frac{1/n}{\frac{1}{2} n^{-1/2}} = \lim_{n\to\infty} \frac{2\sqrt{n}}{n} = \lim_{n\to\infty} \frac{2}{\sqrt{n}} = \emptyset$$

derivata

derivata

ESERCIZIO PER CASA

1)

$f(n) + o(f(n)) \overset{?}{=} \Theta(f(n))$

Come nel polinomio, c'è il termine che prevale ($f(n)$) e gli altri sono o-piccoli

$\Theta(f(n)) \to \lim_{n\to\infty} f(n) = L \qquad \emptyset < L < +\infty$

$f(n) + o(f(n)) = f(n) + \emptyset = f(n)$ OK

2)

$\log n + \sqrt{n} \overset{?}{=} \Theta(\log(n))$

$$\lim_{n\to\infty} \frac{\cancel{\log n}\ 1}{\cancel{\log n}\ 1} + \frac{\sqrt{n}}{\log(n)} = 1 + \text{qualcosa} = L \to \emptyset < L < +\infty$$

ESERCIZIO

$n = O(n \log\log n)$

Cosa succede se considero $n^{1+\varepsilon} = o(n\log\log n) \quad \varepsilon > \emptyset$?

$n\log\log n = O(n^{1+\varepsilon})$

$$\lim_{n\to\infty} \frac{n\log\log n}{n^{1+\varepsilon}} = \lim_{n\to\infty} \frac{\cancel{n}\log\log n}{\cancel{n}\cdot n^{\varepsilon}} \overset{\text{Hôpital}}{=} \lim_{n\to\infty} \frac{\frac{1}{n}\cdot\frac{1}{\log n}}{\varepsilon n^{\varepsilon-1}} =$$

derivata

derivata

$$= \lim_{n\to\infty} \frac{1}{\varepsilon} \cdot \frac{1}{n^{\varepsilon}\log n} = \emptyset = O(n^{1+\varepsilon})$$

$O(1) \Rightarrow f(n) = c \qquad$ cioè: O-grande di 1 = costante

PER CASA

Ordina secondo O-grande:

n^2, $n\log n$, $n^3 + \log n$, $\sqrt{n}$, $n^2 + 2n\log n$, $\log\log n$, $17\log n$, $10n^{3/2}$, $n^5 - n^4 + 2n$, $5n^2\log\log n$, $3n^2 + n^3\log n$, $n + 6\log n$

SOLUZIONE (DAL PIÙ GRANDE AL PIÙ PICCOLO)

$n^5 - n^4 + 2n$, $3n^2 + n^3\log n$, $10n^{3/2}$, $n^3 + \log n$, $n + 6\log n$, n^2, $2n\log n$, $\sqrt{n}$, $17\log n$, $n\log n$, $\log\log n$, $5n^2\log\log n$.

3.10.13

```
RicercaSeq (lista L, elem x)
      for each y ∈ L do
          if (y = x) then return TRUE
          return FALSE
```

$T best (n) = O(1) = Costante$ ⟵ TEMPO MIGLIORE (NON DIPENDE DALLA DIM. DEL DATO IN INGRE

$T worst(n) = c \cdot n = costante \cdot n = O(n)$

$T medio_{(avg)} (n) = c \cdot \frac{n}{2} = O(n)$ ⇝ bisognerebbe conoscere la distribuzione

Ci poniamo sempre nel caso peggiore, con T_{worst}.

$T worst (n) = \max_{istanze I dim (n)} T(I)$

$T best (n) = \min_{istanze I dim(n)}$

$T avg = \sum_{\substack{istanze \\ di dim n}} \varphi(I) \cdot T(I)$

↳ Probabilità di istanza

ALGORITMO

1) Operazione logico-aritmetica /assegnazione ≡ O(1) [costante]

2) if (test) then Body1 else Body2

Tempo di esecuzione di un if = $T(test) + \underbrace{\max\{T(B1) + T(B2)\}}$

↳ il peggiore tra i t_{exe} di B1 e B2

3) for i = 1 to n

Body

$\sum_{i=1}^{n} T_i$ ⟶ t_{exe} corpo

4) while (test) do

Body

5) repeat

Body

until (Test)

$\sum_{i=\emptyset}^{m} T_i' + \sum_{i=\emptyset}^{n-1} T_i$

6) funz (x)

7) seq di istr. I1

I2

I3

Fattoriale (n)

```
ris = 1
for i = 1 to n do
ris = ris * 1
return ris
```

$$T(n) = c_1 + \sum_{i=1}^{n} c_2 = c_1 + c_2 n = \Theta(n)$$

PER CASA

Scrivere un algoritmo in pseudocodice che prenda la lista |3|5|4|6|3|2|9| e determini il massimo locale (maggiore di precedente e successivo) LOCAL MAX

SVOLGIMENTO

```
LocalMax (lista L)
    L = 3, 5, 4, 6, 3, 2, 9
    while (L != 'Ø')
        if (L > L-1 && L < L+1)
        max = L;
        else
        if (L-1 > L)
        max = L-1;
        else
        max = L+1;
        return max;
```

ALGORITMI RICORSIVI

RICORRENZA

$$T(n) = \begin{cases} 1 & n = 1 \\ c + T\left(\frac{n}{2}\right) & n \geq 2 \end{cases}$$

METODO DELL'ITERAZIONE

$$T(n) = c \cdot T\left(\frac{n}{2}\right)$$

$$= c + c + T\left(\frac{n}{2^2}\right) = 2c + T\left(\frac{n}{2^2}\right) = 2c + \left[c + T\left(\frac{n}{2^3}\right)\right] = 3c + T\left(\frac{n}{2^3}\right)$$

$$\to kc + T\left(\frac{n}{2^k}\right) \to k = \log_2 n$$

$$c \cdot \log_2 n + 1 = \Theta(\log n)$$

$$T(n) = \begin{cases} 9\,T\left(\frac{n}{3}\right) + n & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

$$T(n) = 9\,T\left(\frac{n}{3}\right) + n$$

$$= 9\left[9\,T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right] = 9^2\,T\left(\frac{n}{3^2}\right) + (3n + n)$$

$$= 9^3 \cdot T\left(\frac{n}{3^3}\right) + \left(3^2 n + 3n + n\right) = 9^3 \cdot T\left(\frac{n}{3^3}\right) + \left(3^2 n + 3^1 n + 3^0 n\right)$$

$$T(n) = 9^k T\left(\frac{n}{3^k}\right) + n \sum_{i=0}^{k-1} 3^i$$ progressione geometrica

$$= 9^k T\left(\frac{n}{3^k}\right) + n\,\frac{3^k-1}{2}$$

Impongo $\frac{n}{3^k} = 1$, $k = \log_3 n$

$$T(n) = 9^{\log_3 n} + n\,\frac{3^{\log_3 n}-1}{n} =$$

$$= 3^{2\log_3 n} + n\,\frac{n-1}{2} = 3^{\log_3 n^2} + n\,\frac{n-1}{2} = n^2 + n\,\frac{n-1}{2} = \Theta(n^2)$$

9.10.13

ABBIAMO VISTO

1) Metodo dell'iterazione
2) Metodo di sostituzione

ESEMPIO

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ T(\lfloor \frac{n}{2} \rfloor) + n & \text{se } n > 1 \end{cases}$$

$T(n) = O(n)$? $\exists\, c > 0$ tale che $T(n) \le c \cdot n$

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$$

$$\le c \left\lfloor \frac{n}{2} \right\rfloor + n \le c \cdot \frac{n}{2} + n = \left(\frac{c}{2} + 1\right) n \overset{?}{\le} c \cdot n$$

cerco $\frac{c}{2} + 1 \le c \longrightarrow c \ge 2$

DIVIDE ET IMPERA

$$T(n) = \underbrace{T_D(n)}_{\text{DIVIDE}} + \sum_{i=1}^{a} T(n_i) + \underbrace{T_C(n)}_{\text{COMBINE}}$$

(★)

$$\boxed{T(n) = \begin{cases} a\,T\left(\frac{n}{b}\right) + f(n) & n > 1 \\ 1 & n = 1 \end{cases}}$$

$f(n) \longrightarrow T_D(n) + T_C(n)$

$a \ge 1$
$b > 1$
$d = \log_b a$

TEOREMA MASTER

Sia $T(n)$ definita come in (★), con $a \ge 1$, $b > 1$, $f(n)$ asintoticamente non negativa.

1) $T(n) = \Theta(n^d)$, se $f(n) = O(n^{d-\varepsilon})$, per $\varepsilon > 0$

2) $T(n) = \Theta(n^d \log n)$, se $f(n) = \Theta(n^d)$

3) $T(n) = \Theta(f(n))$, se
   - 3.1 $f(n) = \Omega(n^{d+\varepsilon})$, $\varepsilon > 0$
   - 3.2 $\exists\, c < 1$ t.c. $a f(\frac{n}{b}) \le c f(n)$ per n suffic. grande

ESEMPIO 1

$T(n) = 3T\left(\frac{n}{2}\right) + n^2$

Siamo nelle condizioni di usare master?

$a = 3$

$b = 2$ Si

$d = \log_2 3$

$f(n) = n^2$

Ora devo vedere se è $O - \Theta - \Omega$

$n^2 = \Omega(\underbrace{n^{\log_2 3 + \varepsilon}}_{♡})$ $\varepsilon > \emptyset$ $\rightarrow$ verificato in 3.1

$\emptyset < \varepsilon \leq 2 - \log_2 3$ rende vero ♡

$\downarrow$

$\log_2 3 + \varepsilon \leq 2$

$\downarrow$

$n^2 = \Omega(n^{\log_2 3 + \varepsilon})$

$\exists\ c < 1$ t.c. $3\left(\frac{n}{2}\right)^2 \leq c \cdot n^2$? $\rightarrow$ verificato in 3.2

$3\left(\frac{n}{2}\right)^2 \leq c \cdot n^2$

$\hookrightarrow \frac{3}{4}\cancel{n^2} \leq c\cancel{n^2} \rightarrow \frac{3}{4} \leq c < 1$

Quindi $T(n) = \Theta(f(n))$

ESEMPIO 2

$T(n) = 4T\left(\frac{n}{2}\right) + n^2$ $a = 4$ $b = 2$ $d = \log_2 4 = 2$ $f(n) = n^2$

$n^d = n^2 \rightarrow$ caso 2 $\rightarrow T(n) = \Theta(n^2 \log n)$

ESEMPIO 3

$T(n) = T\left(\frac{n}{2}\right) + 2^n$ $a = 1$ $b = 2$ $d = \log_2 1 = \emptyset$ $f(n) = 2^n$

devo confrontare 2^n con $n^\emptyset$

$\varepsilon > \emptyset$. $f(n) = 2^n = \Omega(n^\varepsilon)$

$2^n = \Omega(n)$ se scelgo $\varepsilon = 1$

$\hookrightarrow$ ho che $\Omega(n^{0+1}) = \Omega(n)$

[LA DEF. DICE $\Omega(n^{d+\varepsilon})$]

$\exists\ c < 1$ t.c. $af\left(\frac{n}{b}\right) \leq c \cdot f(n)$?

$\cancel{2}^{\frac{n}{2}} \leq c \cdot 2^n = c \cdot \cancel{2}^{\frac{n}{2}} \cdot 2^{\frac{n}{2}}$

$\hookrightarrow 1 \leq c \cdot 2^{\frac{n}{2}}$

$\hookrightarrow \frac{1}{2^{\frac{n}{2}}} \leq c < 1$

concludo che è il caso 3

ESEMPIO

$$T(n) = 2^n T\left(\frac{n}{2}\right) + n^2$$

$a = 2^n$
$b = 2$
$d = \log_2 2^n = n$

$\rightarrow$ NO TEOREMA MASTER PERCHÈ a NON È COSTANTE!

ESEMPIO

$$T(n) = 16\, T\left(\frac{n}{4}\right) + n$$

$a = 16$ $\quad f(n) = n$
$b = 4$
$d = \log_4 16 = 2$

$\varepsilon > \emptyset$ t.c. $n = O(n^{2-\varepsilon})$ se prendo $\varepsilon = 1$ ottengo: $n = O(n^1) \rightarrow n = O(n)$

primo caso, $T(n) = \Theta(n^2)$

ESEMPIO

$$T(n) = \frac{1}{2} T\left(\frac{n}{2}\right) + \frac{1}{\ }$$

$a = 1/2$ $\rightsquigarrow$ $a < 1$ quindi no MASTER
$b = 2$
$d = \log_2 1/2 = -1$

ESEMPIO

$$T(n) = 2T\left(\frac{n}{2}\right) + n\log n$$

$a = 2$
$b = 2$
$d = \log_2 2 = 1$

Dimostro che non si può usare master perché non si verifica nessuna delle opzioni

1) $n\log n = O(n^{1-\varepsilon})$ $\quad \varepsilon > \emptyset$?

$c > \emptyset$ per n sufficientemente grande $n\log n \leq c \cdot n^{1-\varepsilon} = \frac{n}{n^\varepsilon} \rightarrow n^\varepsilon \log n \leq c$

2) $n\log n = \Theta(n^d) \rightarrow n\log n = \Theta(n)$

$n\log n = O(n)$?

$\lim_{n\to\infty} \frac{\cancel{n}\log n}{\cancel{n}} = +\infty$ Quindi non è mai $O(n)$, quindi non vale

3) $n\log n = \Omega(n^{1+\varepsilon})$ per $\varepsilon > \emptyset$

$$\lim_{n\to\infty} \frac{n\log n}{n^{1+\varepsilon}} = \lim_{n\to\infty} \frac{\cancel{n}\log n}{\cancel{n}\cdot \underbrace{n^\varepsilon}_{\text{prevale}}} = \emptyset$$

Quindi $n\log n = o(n^{1+\varepsilon})$. o-piccolo e Ω-grande sono disgiunti. Quindi non vale.

ESERCIZI PER CASA

① $T(n) = 2T\left(\frac{n}{4}\right) + n^{0.51}$

② $T(n) = 3T\left(\frac{n}{3}\right) + \sqrt{n}$

③ $T(n) = 64T\left(\frac{n}{8}\right) - n^2\log n$

RISOLVO ESERCIZI PER CASA

1) $T(n) = 2T\left(\frac{n}{4}\right) + n^{0.51}$ $a = 2$ $b = 4$ $d = \log_4 2 = 1/2$ $f(n) = n^{0.51}$

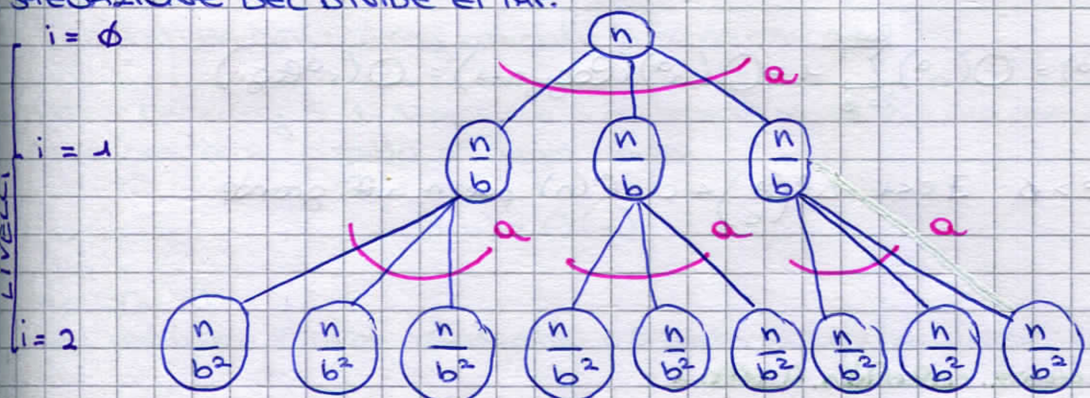
2) $T(n) = 3T\left(\frac{n}{3}\right) + \sqrt{n}$ $a = 3$ $b = 3$ $d = \log_3 3 = 1$ $f(n) = \sqrt{n}$

3) $T(n) = 64T\left(\frac{n}{8}\right) - \underline{n^2 \log n}$ $a = 64$ $b = 8$ $d = \log_8 64 = 2$ $f(n) = \Theta\, n^2 \log n$

↳ non posso applicare il teorema master.

SPIEGAZIONE DEL DIVIDE ET IMP.

10.10.13



1) a livello i la dimensione dei sottoproblemi è $\frac{n}{b^i}$, $i = \emptyset, 1 \ldots$

2) il contributo di un nodo a livello i sul tempo di esecuzione è $f\left(\frac{n}{b^i}\right)$

3) il numero massimo di livello dell'albero è $\log_b n$

4) a livello i ci sono a^i nodi $a^{\log_b n} = n^{\log_b a}$

5) il numero di foglie è n^d ($d = \log_b a$)

$$V(n) = \sum_{i=\emptyset}^{\log_b n} a^i$$

$$T(n) = \sum_{i=\emptyset}^{\log_b n} a^i f\left(\frac{n}{b^i}\right)$$

DIMOSTRAZIONE DEL TEOREMA MASTER

1) $f(n) = O(n^{d-\varepsilon})$, $\varepsilon > 0$ $\Rightarrow T(n) = \Theta(n^d)$

dimostro.

$$a^i f\left(\frac{n}{b^i}\right) = a^i O\left(\left(\frac{n}{b^i}\right)^{d-\varepsilon}\right)$$

$$= O\left(a^i \left(\frac{n}{b^i}\right)^{d-\varepsilon}\right)$$

$$= O\left(n^{d-\varepsilon} \cdot \frac{a^i}{(b^i)^d (b^i)^{-\varepsilon}}\right) = O\left(n^{d-\varepsilon} \cdot \frac{\cancel{a^i} \cdot (b^\varepsilon)^i}{\cancel{a^i}}\right) = O\left(n^{d-\varepsilon} \cdot (b^\varepsilon)^i\right)$$

$$T(n) = \sum_{i=0}^{\log_b n} O\left(n^{d-\varepsilon} \cdot (b^\varepsilon)^i\right) = O\left(\sum_{i=0}^{\log_b n} n^{d-\varepsilon} \cdot (b^\varepsilon)^i\right) = O\left(n^{(d-\varepsilon)} \sum_{i=0}^{\log_b n} (b^\varepsilon)^i\right) =$$

$$= O\left(n^{d-\varepsilon} \cdot \frac{(b^\varepsilon)^{\log_b n+1} - 1}{b^{\varepsilon-1}}\right) = O\left(n^{d-\varepsilon} \frac{(b^\varepsilon)^{\log_b n} \cdot b^{\varepsilon-1}}{b^{\varepsilon-1}}\right)$$

$$T(n) \geq n^d \Rightarrow T(n) = \Omega(n^d)$$

2) $T(n) = \Theta(n^d) \Rightarrow T(n) = \Theta(n^d \log n)$

dimostro.

$$a^i f\left(\frac{n}{b^i}\right) = \Theta\left(a^i \left(\frac{n}{b^i}\right)^d\right) = \Theta\left(n^d \frac{a^i}{(b^i)^d}\right) = \Theta\left(n^d \frac{\cancel{a^i}}{(\cancel{b^d})^i}\right) = \Theta(n^d)$$

$$T(n) = \sum_{i=0}^{\log_b n} \Theta(n^d) = \Theta(n^d) \sum_{i=0}^{\log_b n} 1 = \Theta(n^d)(\log_b n+1) = \Theta(n^d \log n)$$

3) $f(n) = \Omega(n^{d+\varepsilon})$, $\varepsilon > 0$ $\exists\, c<1$ $af\left(\frac{n}{b}\right) \leq c\, f(n)$ per n suff. grande
$\Rightarrow T(n) = \Theta(f(n))$

dimostro. Parte a.

Tempo di esecuz. calcolata a livello 1

$$af\left(\frac{n}{b}\right) \leq c \cdot f(n) < f(n)$$

$$\forall i: \ a^i f\left(\frac{n}{b^i}\right) \leq c f(n)$$

induzione. suppongo vero fino a $(i-1)$

$$a^i f\left(\frac{n}{b^i}\right) = a \cdot a^{i-1} \cdot f\left(\frac{n/b}{b^{i-1}}\right) \leq a \cdot c^{i-1} f\left(\frac{n}{b}\right) = c^{i-1} a f\left(\frac{n}{b}\right) \leq c^{i-1} \cdot c f(n) = c^i f(n)$$

∞ → $c<1$, SERIE GEOMETRICA

$$T(n) \leq \sum_{i=0}^{\log_b n} c^i f(n) = f(n) \sum_{i=0}^{\log_b n} c^i \leq f(n) \sum_{i=0}^{\infty} c^i = f(n) \frac{1}{1-c} = T(n) = O(f(n))$$

→ sono tutti > 0

Parte b. $T(n) = \Omega(f(n))$
$T(n) \geq f(n)$