14 Novembre 8.30 1° compitino, 2° → stesso giorno del totale

1 compitino é recuperabile. ? Febbraio.

Compitini → teoria + esercizi

↓

Sopra lo "0" é "superato" (50% risposte ok)

CONTENUTI CORSO:

Longest common subsequence

↑

INPUT: sequenze di caratteri (LCS), insiemi, grafi (cammini minimi)

↑

Prog. dinamica:

↓

problemi di ottimo, confronto stringhe, cammini minimi.

* LOGICA DI UN ALGORITMO DINAMICO:

1 - Definire ricorsivamente il problema → ma non si usa alg. ricors. puro

2 - Si usa algoritmo iter. bottom-up sfruttando la regola ricorsiva.

* ALTRI ARGOMENTI (piú MARGINALI)

Algoritmi Greedy, Esplorazione di un grafo.

RIPASSO (continua):

Ricorrenza merge_sort( ) ⟶ ultima lezione (riprende)

CON "SROTOLAMENTO":

$*$ Strategia più easy: TEOREMA DELL' ESPERTO

$T(n) = 2\,T(n/2) + \Theta(n)$

Supponiamo che n sia una potenza di 2 e che $\Theta(n) = n$.

ALBERO CHIAMATE:

⟶ tempo "merge" × premesse

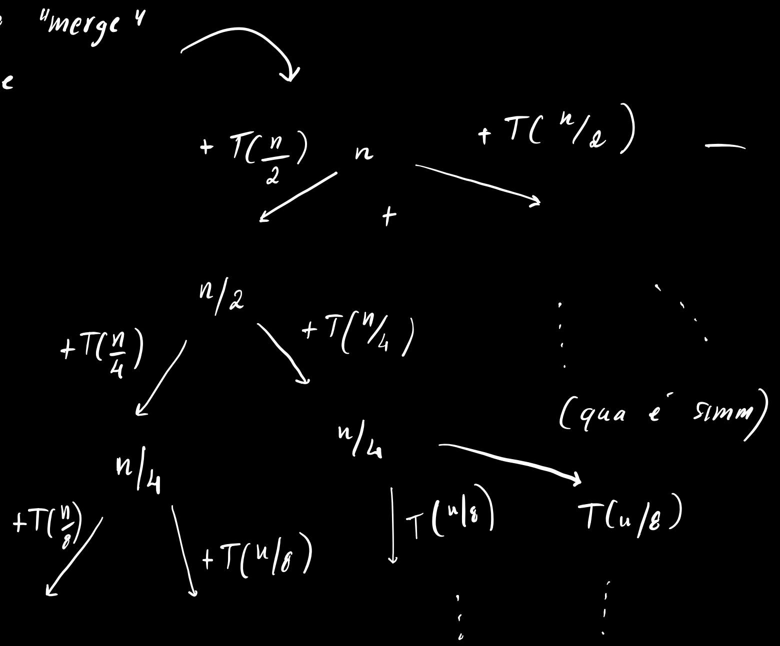
$$T(n) = \underbrace{2\,T\left(\frac{n}{2}\right)}_{\text{tempo "divide"}} + \underbrace{n}_{\text{tempo "merge" x premesse}}$$

$T\left(\frac{n}{2}\right) = 2\,T\left(\frac{n}{4}\right) + \frac{n}{2}$

$T\left(\frac{n}{4}\right) = 2\,T\left(\frac{n}{2}\right) + \frac{n}{4}$

$T\left(\frac{n}{8}\right) = 2\,T\left(\frac{n}{4}\right) + \frac{n}{4}$

....



Arrivo al caso base

$T(1) = 1$

Su ogni livello di sottochiamate → n = somma $T(n)$ sottochiamate.

{ quindi il tempo complessivo,
per $k+1$ livelli ho:
↓
liv. 0

$T(n) = (k+1) \cdot n$

Successioni livelli: $n \rightarrow \frac{n}{2} \rightarrow \frac{n}{2^2} \rightarrow \dots \rightarrow \frac{n}{2^k} = n$

liv: 0 1 2

ALBERO "PIENO"

$T(n) = (\log(n) + 1) \cdot n = \Theta(n \log n)$

$\frac{n}{2^k} = n \to \boxed{k = \log(n)}$

METODO DELL' ESPERTO (fondato sul discorso precedente)

Se ho una ricorrenza del tipo:

$$T(n) = a \cdot T(n/b) + f(n)$$

con $a \geq 1$; $b > 1$ e $f(n)$ definit. pos.

1. Se $\exists \varepsilon > 0$ t.c. $f(n) = O(n^{\log_b a - \varepsilon})$, allora:

$$T(n) = \Theta(n^{\log_b a})$$

2. Se $f(n) = \Theta(n^{\log_b a})$

$$T(n) = \Theta(n^{\log_b a} \cdot \log(n))$$

3. → RIPASSO PERSONALMENTE.

ESEMPIO: (sempre merge sort) → APPLICAZIONE TEOREMA ESPERTO.

$T(n) = 2\, T\left(\frac{n}{2}\right) + n$ è nella forma:

$T(n) = a\, T\left(\frac{n}{b}\right) + f(n)$

$a = 2$; $b = 2$ $\qquad \log_b a = \log_2 2 = 1$

1. → $f(n) = n$ è $O(n^{1-\varepsilon})$? NOPE

2. → $f(n) = n$ è $\Theta(n^1)$? SÌ, PERCHÉ n^1 È INF. LIM. DA $n^{1-\varepsilon}$

$\Rightarrow T(n) = \Theta(n \cdot \log(n))$ → ESATTO!

ES: $\log_2^2 = 1$ → primo caso perché

→ se uso $1/4$ ad esempio

1. $5 = O(n^{1-\varepsilon})$

$\exists \varepsilon$? yes ergo $T(n) = \Theta(n)$

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

$a = 1$ $b = 2$ $\log_2^1 = 0$

quindi: $1 = O(n^{0-\varepsilon})$ → No, $\underline{\varepsilon > 0}$ CASO 1 NON FUNZ.

$1 = \Theta(n^0)$ → sì → $\Theta(n^0 \cdot \log n) = \Theta(\log(n))$

FIBONACCI (Calcolo complessità):

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

$\boxed{\text{fib\_r :}}$

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

* COMPLESSITÀ (COMPATTO):

$T(n) \longrightarrow$ FIB_R(n)

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

ritorno, controllo

| ESISTE UN ALGO IT. FUNZIONANTE DI FIBO. QUESTO (RICORS) È DIDATTICO × INTRODURRE PROG. DINAMICA.

FIB_IT (n)

```
x = 0
y = 1
for i=2 to n
    z = x + y
    x = y
    y = z
return (x o z, non ricordo).
```

SROTOLAMENTO (della funzione FIBO):

$T(n) = T(n-1) + T(n-2) + 2$ → "trascurabile"

CALCOLO r (in r^n) → (base tempo esponenziale)

$r^n = r^{n-1} + r^{n-2} \dots + r$

* moltiplico per r^2 e divido r^n

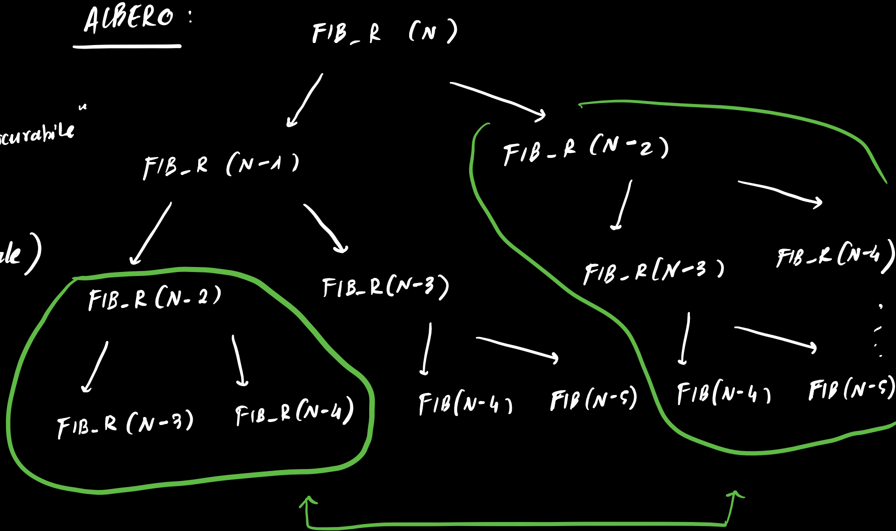
$$r^2 = \frac{r^{n-1} + r^{n-2} \dots r \cdot r^2}{r^n}$$

$r^2 = r + 1$

$r^2 - r - 1 = 0 \rightarrow r = \frac{1 \pm \sqrt{5}}{2}$ → quindi $T(n) = \left(\frac{1+\sqrt{5}}{2}\right)^n$ opp. $T(n) = \left(\frac{1-\sqrt{5}}{2}\right)^n$

oppure $T(n) = c_1 \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + c_2 \left(\frac{1-\sqrt{5}}{2}\right)^n - \boxed{2}$

ALBERO:



Sono ripetuti (molti calcoli → più volte)

2 o + invocazioni fanno lo stesso lavoro.

↓

Complessità computazionale terribile: ESPONENZIALE r^n

perciò → eq. lin. → sp. $\vec{v}$ → BASE → soluz = $\gamma_1 b_1 + \gamma_2 b_2$

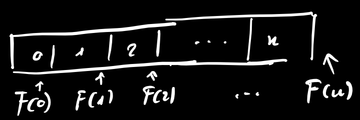
Con le condizioni iniziali:

$T(0) = 2$ e $T(1) = 2$ } tra le $c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$ combinazioni piglio quelle con $n=1$ e $n=0$

$\Theta\left[\left(\frac{1+\sqrt{5}}{2}\right)^n\right]$ complessità computazionale

$\frac{1+\sqrt{5}}{2}$ è sezione Aurea → FIBO! cercalo in giro.

Come risolviamo → Salvo valori parziali delle ricorsioni



$F[0] = 0$

$F[1] = 1$

for $i = 2$ to n

$F[i] = F[i-1] + F[i-2]$

↓ celle dell'array

è iterativo → $\Theta(n)$

↓

"Scimmiotta" Algo iter.

Problema Hateville