Algoritmi 3

Prof. Domenico Cantone

A.A. 2004-2005

PROGRAMMA

- ANALISI AMMORTIZZATA
- ALGORITMI SU GRAFI
  - ALBERI RICOPRENTI MINIMI
  - RETI DI FLUSSO
- STRUTTURE DATI AVANZATE
  - HEAP BINOMIALI
  - HEAP DI FIBONACCI
  - SPLAY TREES
- PROBLEMI DI STRING MATCHING
- ALGORITMI PARALLELI

ANALISI AMMORTIZZATA

- PER ANALIZZARE SEQUENZE DI n OPERAZIONI
- SI DETERMINA UN TEMPO COMPLESSIVO $T(n)$ CHE VIENE RIPARTITO IN QUALCHE MODO TRA LE n OPERAZIONI
- ES. $T(n)/n$ - COSTO AMMORTIZZATO PER OPERAZIONE
- LA STIMA OTTENUTA NON E' PROBABILISTICA, MA SI TRATTA DI UNA MEDIA NEL CASO PEGGIORE

TRE METODI:

- METODO DELL'AGGREGAZIONE
- METODO DEGLI ACCANTONAMENTI
- METODO DEL POTENZIALE

DUE ESEMPI:

- STACK CON MULTIPOP
- CONTATORE BINARIO CON INCREMENT

TABELLE DINAMICHE

STACK CON MULTIPOP

POP(S) → COSTO $O(1)$

PUSH(S, x) → COSTO $O(1)$

STACK_EMPTY(S) → COSTO $O(1)$

```
MULTIPOP (S, k)
  while not STACK_EMPTY(S) and k ≠ 0 do
      POP(S)
      k := k - 1
```

MULTIPOP (S, k) → COSTO $O(\min(|S|, k))$

ANALISI DI UNA SEQUENZA DI n OPERAZIONI SU UNO STACK INIZIALMENTE VUOTO

- $|S| = O(n)$
- COSTO DI UNA SINGOLA OPERAZIONE $= O(n)$
- COSTO DI n OPERAZIONI $= nO(n) = O(n^2)$

CONTATORE BINARIO CON INCREMENT

- SIA $A[0..k-1]$ UN ARRAY DI k BIT

$$VALUE[A] = \sum_{i=0}^{k-1} A[i] \cdot 2^i$$

$$VALUE[INCREMENT(A)] \equiv VALUE[A] + 1 \pmod{2^k}$$

$1101011 \longmapsto 1101100$

```
INCREMENT(A)
  i := 0
  while i<k and A[i]=1 do
      A[i] := 0
      i := i+1
  if i<k then
      A[i] := 1
```

					$\downarrow i$
0	1	~~0~~	~~1~~	~~1~~	~~1~~
5	4	3	2	1	0
		1	0	0	0

COSTO DI UN INCREMENTO $= O(k)$

COSTO DI n INCREMENTI $= nO(k) = O(nk)$

METODO DELL'AGGREGAZIONE

- CONSISTE NELLO STIMARE IL COSTO $T(n)$ DI n OPERAZIONI E DI EQUIDISTRIBUIRE TALE COSTO TRA LE n OPERAZIONI $\left(T(n)/n \right)$

STACK CON MULTIPOP (INIZIALMENTE VUOTO)

$$\left.\begin{array}{l} POP(S) \\ PUSH(S,x) \end{array}\right\}$$ OPERAZIONI ELEMENTARI

MULTIPOP (S, k) – OPERAZIONE DERIVATA

$$\underbrace{op_1, op_2, op_3, \dots, op_{n-1}, op_n}_{\displaystyle\downarrow} \in \{POP, PUSH, MULTIPOP\}$$

$$op'_1, op'_2, op'_3, \dots, op'_{m-1}, op'_m \in \{POP, PUSH\}$$

$$COSTO(\langle op_1 .. op_n \rangle) = COSTO(\langle op'_1 .. op'_m \rangle) = m$$

$$\#POP(\langle op'_1 .. op'_m \rangle) \leq \#PUSH(\langle op'_1 .. op'_m \rangle)$$

$$\#PUSH(\langle op'_1 .. op'_m \rangle) = \#PUSH(\langle op_1 .. op_m \rangle) \leq n$$

PERTANTO:

$$\#POP(\langle op'_1 .. op'_m \rangle) \leq n$$

DA CUI

$$COSTO(\langle op_1 .. op_m \rangle) = COSTO(\langle op'_1 .. op'_m \rangle)$$
$$= \#POP(\langle op'_1 .. op'_m \rangle) + \#PUSH(\langle op'_1 .. op'_m \rangle)$$
$$\leq n + n = 2n$$

COSTO_AMMORTIZZATO_PER_OPERAZIONE ≤ 2

CONTATORE BINARIO CON INCREMENT (INIZIALMENTE NULLO)

OPERAZIONI ELEMENTARI: SET E RESET DI SINGOLI BIT

..00000
..00001
..00010
..00011
..00100
..00101
..00110
..00111
..01000
..01001
..01010

SU n OPERAZIONI INCREMENT

A[0] CAMBIA n VOLTE

A[1] CAMBIA $\left\lfloor \frac{n}{2} \right\rfloor$ VOLTE

A[2] CAMBIA $\left\lfloor \frac{n}{2^2} \right\rfloor$ VOLTE

A[3] CAMBIA $\left\lfloor \frac{n}{2^3} \right\rfloor$ VOLTE

...

$$T(n) = \sum_{i=0}^{\lfloor \lg n \rfloor} \left\lfloor \frac{n}{2^i} \right\rfloor \leqslant \sum_{i=0}^{\lfloor \lg n \rfloor} \frac{n}{2^i} < \sum_{i=0}^{\infty} \frac{n}{2^i} = 2n$$

COSTO_AMMORTIZZATO_PER_OPERAZIONE ≤ 2

METODO DEGLI ACCANTONAMENTI

$op_1, op_2, \ldots, op_n$

$c_i =_{df}$ COSTO_REALE (op_i)

$\hat{c}_i =_{df}$ COSTO_AMMORTIZZATO (op_i) (DEFINITO DA NOI)

OBIETTIVO

DEFINIRE I COSTI AMMORTIZZATI IN MODO TALE CHE VALGA

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i$$

METODO DEGLI ACCANTONAMENTI (CNT)

SE $\hat{c}_i > c_i$, c_i UNITÀ DI COSTO SONO UTILIZZATE PER PAGARE IL COSTO DI op_i

$\hat{c}_i - c_i$ UNITÀ DI COSTO SONO IMMAGAZZINATE SU ELEMENTI SPECIFICI DELLA STRUTTURA DATI

SE $c_i > \hat{c}_i$, LA DIFFERENZA $c_i - \hat{c}_i$ VIENE RECUPERATA DA CREDITI IMMAGAZZINATI NELLA STRUTTURA DATI

$\therefore$ VIENE RAGGIUNTO L'OBIETTIVO

STACK CON MULTIPOP (INIZIALMENTE VUOTO)

$\hat{c}_{PUSH} = 2$ (1 UNITÀ PER IL COSTO REALE + 1 UNITÀ ASSEGNATA ALL'ELEMENTO)

$\hat{c}_{POP} = \hat{c}_{MULTIPOP} = 0$

IN OGNI ISTANTE: $\sum_{i=1}^{n} \hat{c}_i - \sum_{i=1}^{n} c_i = |S| \geq 0$

E QUINDI $\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$

PERTANTO $\sum_{i=1}^{n} c_i \leq 2n$

CONTATORE BINARIO CON INCREMENT
(INIZIALMENTE NULLO)

$\hat{c}_{SET} = 2$ (1 UNITÀ PER PAGARE L'OPERAZIONE + 1 UNITÀ IMMAGAZZINATA SUL BIT STESSO)

$\hat{c}_{RESET} = 0$

$\hat{c}_{INCREMENT} \leq 2$

$$\sum_{i=1}^{p} \hat{c}_i - \sum_{i=1}^{p} c_i \geq \text{\# BIT SUL CONTATORE UGUALI AD 1} \geq 0$$

$\therefore$ VALE $\displaystyle\sum_{i=1}^{p} \hat{c}_i \geq \sum_{i=1}^{p} c_i$

METODO DEL POTENZIALE

- ALLA STRUTTURA DATI VIENE ASSEGNATA UNA FUNZIONE POTENZIALE

$$D_0 \xrightarrow[op_1]{} D_1 \xrightarrow[op_2]{} D_2 \xrightarrow[op_3]{} D_3 \xrightarrow[op_4]{} \cdots \xrightarrow[op_n]{} D_n$$

DEFINIZIONE

$\phi : D \longmapsto \phi(D) \in \mathbb{R}$ (POTENZIALE)

$op_i \longrightarrow c_i$ (COSTO REALE)

$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$ (COSTO AMMORTIZZATO)

$$\sum_{i=1}^{m} \hat{c}_i = \sum_{i=1}^{m} \left[c_i + (\Phi(D_i) - \Phi(D_{i-1})) \right]$$
$$= \sum_{i=1}^{m} c_i + \Phi(D_m) - \Phi(D_0)$$

LEMMA $\sum_{i=1}^{m} \hat{c}_i \geq \sum_{i=1}^{m} c_i$ SSE $\Phi(D_n) \geq \Phi(D_0)$

$\Phi(D_k) \geq \Phi(D_0)$ per $k = 1, 2, \ldots, n$

STACK CON MULTIPOP (INIZIALMENTE VUOTO)

PONIAMO: $\Phi(S) =_{df} |S|$

SE S_0 È LO STACK VUOTO, $\Phi(S_0) = 0$.

QUINDI $\Phi(S) \geq \Phi(S_0)$.

$$\hat{c}_{POP} = c_{POP} + \Phi(S_i) - \Phi(S_{i-1}) = 1 + |S_i| - |S_{i-1}|$$
$= 0$ IN QUANTO $|S_i| = |S_{i-1}| - 1$

$$\hat{c}_{MULTIPOP} = c_{MULTIPOP} + \Phi(S_i) - \Phi(S_{i-1})$$
$= k + |S_i| - |S_{i-1}| = 0$ (IN QUANTO $|S_i| = |S_{i-1}| - k$)

$$\hat{c}_{PUSH} = c_{PUSH} + |S_i| - |S_{i-1}| = 1 + 1 = 2$$
(IN QUANTO $|S_i| = |S_{i-1}| + 1$)

PERTANTO: $\sum_{i=1}^{m} c_i \leq \sum_{i=1}^{m} \hat{c}_i \leq 2n$

CONTATORE BINARIO CON INCREMENTO (INIZIALMENTE NULLO)

$\Phi(D_i) = \#$ BIT UGUALI AD 1

SIA D_0 LA CONFIGURAZIONE NULLA DEL CONTATORE

$\Phi(D_0) = 0.$

INOLTRE $\Phi(D_i) \geq \Phi(D_0)$

$\hat{c}_{INCREMENT} = c_{INCREMENT} + \Phi(D_i) - \Phi(D_{i-1})$

SI HA:

$c_{INCREMENT} \leq \# RESET_i + 1$

$\Phi(D_i) \leq \Phi(D_{i-1}) - \# RESET_i + 1$

PERTANTO

$$\hat{c}_{INCREMENT} \leq (\# RESET_i + 1) + (\Phi(D_{i-1}) - \# RESET_i + 1) - \Phi(D_{i-1}) = 2$$

DA CUI

$$\sum_{i=1}^{m} c_i \leq \sum_{i=1}^{m} \hat{c}_i \leq 2n$$

CONTATORE BINARIO CON INCREMENTO INIZIALMENTE NON NULLO

PONIAMO COME PRIMA

$$\phi(D_i) = \#\ \text{BIT UGUALI AD } 1$$

SI HA

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \hat{c}_i - \Phi(D_n) + \Phi(D_0)$$

$$\leq 2n + \Phi(D_0) \leq 2n + k$$

(DOVE k E' IL NUMERO DI BIT DEL CONTATORE)

SE $k = O(n)$ ALLORA

$$\sum_{i=1}^{n} c_i = O(n)$$

ESERCIZI

1) SIMULARE UNA CODA CON DUE STACK ED ANALIZZARE LA SIMULAZIONE CON L'ANALISI AMMORTIZZATA

2) ANALIZZARE IL CONTATORE BINARIO ANCHE IN PRESENZA DELL'OPERAZIONE RESET CHE HA L'EFFETTO DI PORRE A ZERO TUTTI I BIT DEL CONTATORE