Lezione 23 aprile

Thursday, April 23, 2020 9:08 AM

Tabelle hash

- inserzione
- ricerca
- cancellazione (liste di trabocco)
  (virtuale con open hash)

Operazioni nel caso pessimo $O(n)$!

nel caso medio $\Theta(1)$

$\alpha = \frac{n}{m}$ numero el. memorizzati / size della tabella

α fattore di carico / di riempimento

Scansione lineare

$$h(k,i) = (h(k,0) + q\,i) \bmod m$$

q sia primo con m

$m = 11 \qquad q = 1$

$$h(k) = h(k,0) = k \bmod m$$

inserire in tabella le chiavi

43, 22, 31, 4, 15, 28, 17, 86, 60

0	1	2	3	4	5	6	7	8	9	10
22	86			4	15	28	17	60	31	43

T

$h(43) = 10 \qquad h(22) = 0 \qquad h(31) = 9$

$h(4) = 4 \qquad h(15) = 4 \qquad h(15,1) = 4 + 1 = 5$

$h(28) = 6 \qquad h(17) = 6 \qquad h(17,1) = 6 + 1 = 7$

$h(86) = 9 \qquad h(86,1) = 10 \qquad h(86,2) = 11 = 0$

$h(86,3) = 9 + 3 = 12 = 1$

$h(60) = 5 \quad \ldots\ldots = 8$

$$\alpha = \frac{9}{11} = 0.82\%$$

<u>Teorema</u>: il numero medio di accessi per ricerca e inserzione è $\frac{1}{1-\alpha}$.

<u>Ipotesi</u>: sequenza di inserzione sia casuale (nella scansione lineare è falso)

quante sequenze di scansione diverse? m

Prova

$T(n, m)$ = il numero medio di accessi per inserire in tabella la chiave ennesima in una tabella di m locazioni.

$T(0, m) = 1$

pr(chiave n-sima trovi $h(k)$ libera) $= \frac{m-n}{m} \to 1$ acc.

pr(........ casella occupata) $= \frac{n}{m}$

$\to$ 1 accesso $+ T(n-1, m-1)$

$$T(n, m) = \frac{m-n}{n} + \frac{n}{m}\left(1 + T(n-1, m-1)\right)$$

$$T(n, m) = \begin{cases} 1 & \text{se } n = 0 \\ 1 + \frac{n}{m} \cdot T(n-1, m-1) & \end{cases}$$

Trial & error

ipotesi: $T(n, m) \leq \frac{m}{m-n}$

dimostriamo per induzione

$n = 0 \quad T(0, m) = 1$

$$T(n,m) \leq 1 + \frac{n}{m} \cdot \left(\frac{m}{m-n}\right) =$$

$$= 1 + \frac{n}{m-n} = \frac{m}{m-n}$$ vero

$$= \frac{1}{1-\frac{n}{m}} \leftarrow \alpha = \frac{1}{1-\alpha}$$

$\alpha = \frac{9}{11}$ n° accessi $= \frac{1}{1-9/11} = \frac{11}{2} = 5.5$ accessi

Altre leggi di scansione

scansione quadratica il salto aumenta in modo quadratico rispetto al tentativo

$$h(K,i) = \left(h(K,0) + a i^2 + b i + c\right) \bmod m$$

per opportuni valori di a, b, c.

L1: $h(K,i) = (h(K,0) + i^2) \bmod m$, m primo

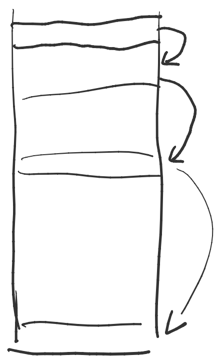
L2: $h(K,i) = \left(h(K,0) + \frac{1}{2} i^2 + \frac{1}{2} i\right) \bmod m$

$m = 2^1$

L1: non colpisce tutte le m posizioni della tabella dopo $\frac{m+1}{2}$ tentativi.

L2: è totale

sequenze di scansione sono solo m.



scansione quadratica tende a sparpagliare di più le chiavi.

<u>Doppio hash</u>: legge di scansione svincolata da indirizzo hash

$$h(K,i) = \left(h(K,0) + i\left(h'(K,0) + 1\right)\right) \bmod m$$

$h(K)$ e $h'(K)$ funzioni hash distinte

L3: m primo

$h(K) = K \bmod m$

$h'(K) = K \bmod (m-1)$

L4: $h(k)$: s bit "estratti" da k;
$h'(k)$: $2b$ bit "estratti" da k con $b = s-1$;

$m = 2^s$

Quante sequenze di scansione si ottengono?

m^2

Esempio di L1 : scansione quadratica

$$h(k,i) = (h(k,0) + i^2) \bmod m$$

$m = 11$

0	1	2	3	4	5	6	7	8	9	10
22		86	60	4	15	28	17		31	43

43, 22, 31, 4, 15, 28, 17, 86, 60

$h(k) = k \bmod m$

$h(k,1) = 4 + i^2 = 5$
$h(k,1) = 6 + i^2 = 7$

$h(86,1) = 9 + 10 = 10$
$h(86,2) = 9 + 4 = 2$

$h(60) = 5$
$h(60,1) = 6$

$h(60,2) = 5 + 4 = 9$
$h(60,3) = 5 + 9 = 14 = 3$

L3 es: $h(k,i) = k \bmod m + i(k \bmod (m-1) + 1)$

$m = 11$

T

22	86		17	4	15	28	60		31	43
0	1	2	3	4	5	6	7	8	9	10

$\underline{43, 22, 31, 4,}$ 15, 28, 17, 86, 60

15
$$\begin{cases} k \bmod m = 4 \\ k \bmod (m-1) = 5 \end{cases}$$

$h(k,1) = 4 + (5+1) = 10$

$h(15,2) = 4 + 2(6) = 16 \bmod 11 = 5$

17

$k \bmod m = 6$

$k \bmod m-1 = 7$

$h(17,1) = 6 + 8 = 14 \bmod 11 = 3$

86

$h(86) = 9$

$h'(86) = 6$

$h(86,1) = 9 + 7 = 5$

$h(86,2) = 9 + 14 = 23 \bmod 11 = 1$

60

$h(60) = 5$

$h'(60) = 0$

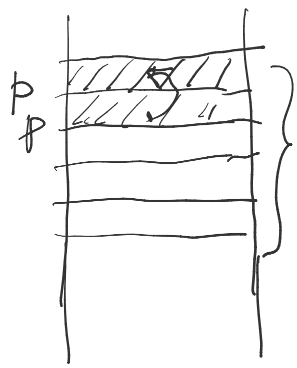
$h(60,1) = 5 + 1 = 6$

$h(60,2) = 5 + 2 = 7$

Scansione lineare passo 1

cancellazione

senza usare la marcatura
ma analizzando l' agglomerato



- esame di tutti gli elementi che seguono nell'agglomerato
- si ricalcola la funzione hash q di questi el.
- se $q \leq p$ si sposta l'el. nella cella p.
- la cella da considerare vuota è quella che si libera

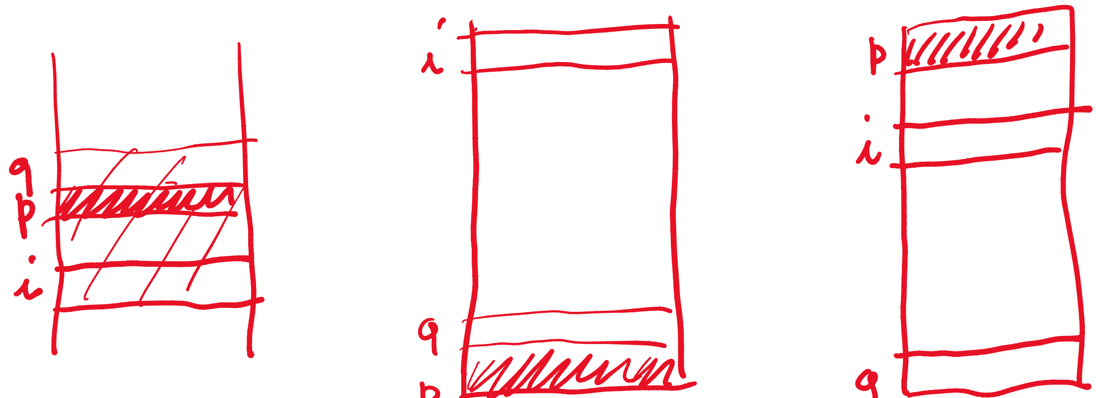
```
HashCanc (A, K):
p = h(K);
while (A[p] != K && A[p] != NULL)
        p = (p+1) % m;
if A[p] == NULL return "K non è presente"
else {      A[p]=K , p posizione vuota
```

ricerca di K

analisi agglomerato

```
i = (p+1) % m;
while (A[i] != NULL { q = h(A[i]);
  if (q ≤ p < i) || (i < q ≤ p) || (p < i < q) {
      A[p] = A[i]; p = i;
  }
  i = (i+1) % m
}
```

condizioni scambio



caso 1