

ESERCIZIO 1

1) La funzione

$$h(k, i) = (14k + i) \bmod 7$$

è utilizzabile, ma produce la stessa sequenza di scansione

$$\langle 0, 1, 2, 3, 4, 5, 6 \rangle$$

per ogni chiave, con conseguente degrado delle prestazioni. Infatti

$$\begin{aligned} h(k, i) &= (14k + i) \bmod 7 = \underbrace{(14k) \bmod 7}_{=0} + i \bmod 7 \bmod 7 \\ &= i \bmod 7 \end{aligned}$$

$\Rightarrow h(k, i)$ dipende da i , ma non dalla chiave k

2) Doppio Hash

$$h_1(k) = k \bmod 7$$

$$h_2(k) = 1 + k \bmod 6$$

$$h(k, i) = (k \bmod 7 + i(1 + k \bmod 6)) \bmod 7$$

$$T = \begin{array}{|c|c|c|c|c|c|c|} \hline 4 & 1 & 2 & & 11 & & 9 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline \end{array}$$

Chiave	h_1	h_2	sequenza
2	2	4	2
9	2		2, 6
1	1	5	1
11	4		4
4	4		4, 2, 0

ESERCIZIO 2

1) $T(n, m) = \Theta(n+m)$

scansione dell'intera tabella ($\Theta(m)$) e di tutte le liste di tabacco ($\Theta(n)$)

2) $T(n, m) = \Theta(m)$

$n \leq m$. La tabella deve essere esaminata interamente perché non sono note a priori le celle occupate.

ESERCIZIO 3

1) chiavi Distinte (S)

D = nuovo dizionario;

c = 0;

```
for (i=0; i<n; i++) {  
    if (search(D, S[i]) == NIL) {  
        insert(D, S[i]);
```

```
    }  
    c++;
```

```
}  
return c;
```

2) D contiene solo gli r elementi distinti di S. Se D è implementato con un array ordinato, ogni ricerca costa $O(\log r)$ [→ Ricerca Binaria] e ogni inserimento costa $O(r)$. [→ per mantenere l'array ordinato]. Dato che le ricerche sono N e gli inserimenti r, il costo complessivo è $O(n \log r + r^2)$

ESERCIZIO 4

CercaESposta(T, k)

$h = \text{hash}(k);$

$v = \text{Search}(T[h], k)$ // ricerca in una lista

if ($v \neq \text{NIL} \ \&\& \ v \neq T[h]$) {

if ($v.\text{next} \neq \text{NIL}$) $v.\text{next}.\text{prev} = v.\text{prev};$

$v.\text{prev}.\text{next} = v.\text{next};$

} $v.\text{prev} \neq \text{NIL}$ perché
 $v \neq T[h]$ dunque v non
è la testa della lista

$v.\text{prev} = \text{NIL};$

$v.\text{next} = T[h];$

$T[h].\text{prev} = v;$

$T[h] = v;$

}

return $v;$

$\Theta(1)$

Il costo è quello di una ricerca in una tabella hash con concatenamento.

$T(n, m) = O(n)$ al caso pessimo

$T(n, m) = \Theta(1 + \alpha)$ al caso medio.

$$\alpha = \frac{n}{m}$$

ESERCIZIO 5

Somma (a, k)

T = nuova tabella hash di dimensione $2 \times n$

for ($i = 1$; $i \leq n$; $i++$) {

$u = \text{SEARCH}(T, k - a[i])$

if ($u == \text{NIL}$) {

$v =$ nuovo elemento per T

$v.\text{key} = a[i];$

$v.\text{indice} = i;$

$\text{INSERT}(T, v);$

}

else // $a[i] + a[u.\text{indice}] = k$ ($u.\text{key} = k - a[i]$)

return $\langle u.\text{indice}, i \rangle;$

}

return $\langle -1, -1 \rangle;$ // \neq coppia di somma k

$T(n) = O\left(n * \text{Costo della Ricerca/Inserimento}\right)$
nella tabella hash T

$= O(n)$

$\Theta(1)$ al caso medio