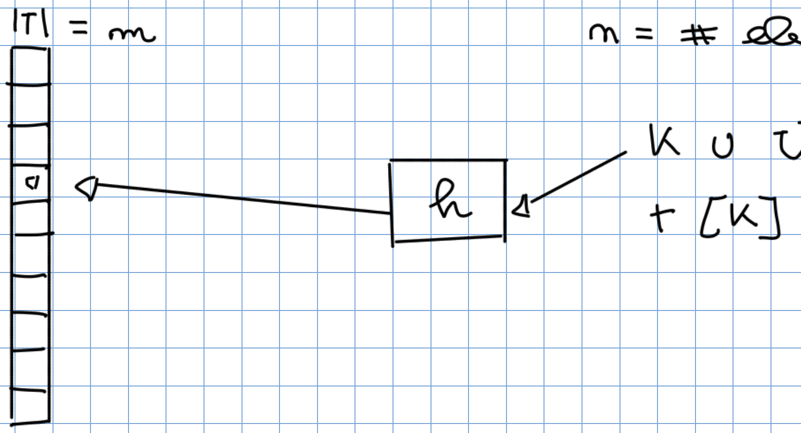
tabelle hash

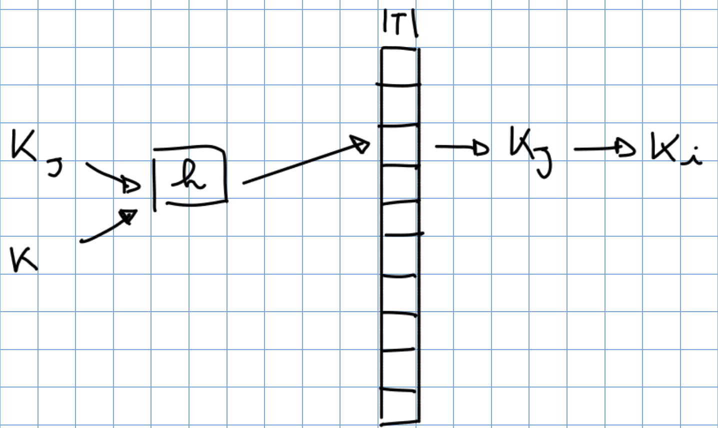
$n = \#$ elementi → numero di elementi inseriti



Collisioni

$K_i, K_j \in U \quad h(K_i) = h(K_j)$

Concatenazione



Posso risolvere il problema delle collisioni considerando le singole celle come delle liste, le funzioni diventano:
→ doppiamente concatenate

Insert(T, K)

  List-insert(K, T[h(K)])

Search(T, K)

  Return List-search(K, T[h(K)])

Questo apre il problema: se tutti collidono nella stessa posizione diventa una lista concatenata con una complessità computazionale in ricerca peggiore. Per capire la nostra situazione nasce il fattore di carico

lunghezza media → $\alpha = \frac{n}{m}$ che indica quanti elementi abbiamo inserito rispetto al numero di celle

Ipotesi di hashing G uniforme semplice

$k \in U$

probabilità

$0 \leq i < m$

$Pr\{h(k) = i\} = \frac{1}{m}$

tutti i valori inseriti nella tabella hanno tutti la stessa probabilità di essere inseriti nella posizione i

Ricerca senza successo → cerchiamo qualcosa che potrebbe non esserci

ha:

- n elementi
- m posizioni

$$|T[i]| = \sum_{i=1}^{n} Pr\{h(k) = i\} = \sum_{i=1}^{n} \frac{1}{m} = \frac{n}{m} = \alpha$$

quindi ha complessità $O(\underline{1} + \alpha)$

↳ accede al null successivo per capire se sono arrivato alla fine della lista

Ricerca con successo → cerchiamo qualcosa che c'è sicuramente

$k_1, k_2, k_3 \dots k_i, k_{i+1} \dots k_n$

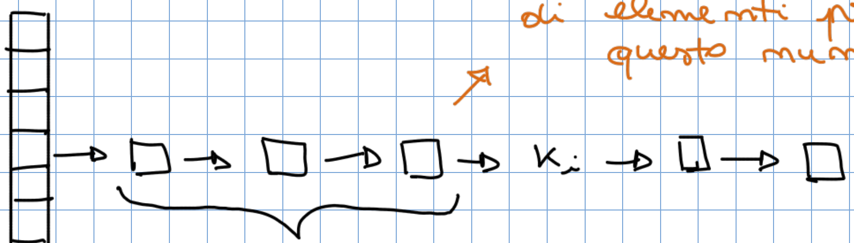
elemento da cercare

ipotesi

$x_{ij} = I\{h(k_i) = h(k_j)\}$

$Pr\{x_{ij}\} = \frac{1}{m}$

la complessità si basa sul numero di elementi prima di k_i, troviamo questo numero facendo



numero di elementi definito come: $1 + \sum_{j=i+1}^{n} Pr\{x_{ij}\}$

il lavoro medio lo troviamo facendo

$$\frac{1}{n}\sum_{i=1}^{n}\left[1+\sum_{j=m+1}^{n}\underbrace{Pr\{X_{ij}\}}_{\frac{1}{m}}\right] = \frac{1}{n}\left(\underbrace{\sum_{i=1}^{n} 1}_{n} + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{1}{m}\right)$$

essendo $\frac{1}{m}$ una costante lo scrivo fuori

$$= \frac{1}{n}\left(n + \frac{1}{m}\sum_{i=1}^{n}\underbrace{\sum_{j=i+1}^{n} 1}_{n-i}\right) = \frac{1}{n}\left(n + \frac{1}{m}\left(\underbrace{\sum_{i=1}^{n} n}_{n^2} - \underbrace{\sum_{i=1}^{n} i}_{\frac{n(n+1)}{2}}\right)\right)$$

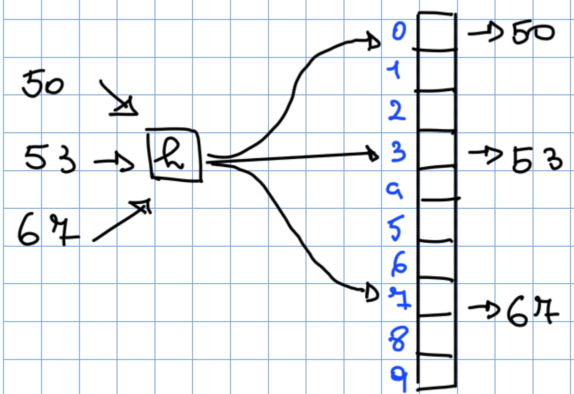
$$= 1 + \frac{1}{nm}\left(n^2 - \frac{n(n+1)}{2}\right) = 1 + \frac{n}{m} - \frac{n+1}{2m} \in O(1+\alpha)$$

tabelle hash - metodo della divisione

$h: U \to \{0, 1, 2, \dots m-1\}$

uno dei metodi per definire h (più noto)

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



$m=10$

usare $m = 2^p$ non è conveniente

tabelle hash - metodo della moltiplicazione

$h: U \to \{0, 1, 2, \dots m-1\}$

$0 < A < 1$

$$h(k) = \left\lfloor m\,[\underbrace{kA}_{0 < kA < k} \bmod 1] \right\rfloor$$

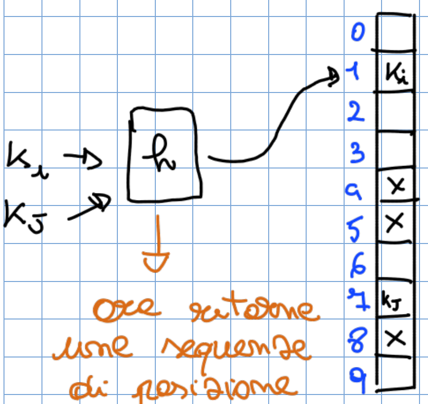
escludo la parte decimale

$\to 0 \le kA \bmod 1 < 1$

$0 \le m\,(kA \bmod 1) < m$

prendo il floor perché mi serve la parte intera

Tabelle hash - indirizzamento aperto



ora ritorna una sequenza di posizione

$h(k_i) = h(k_J)$

$h(k,0)$
$h(k,1)$
$h(k,2)$
$\vdots$
$h(k,i)$

tentativi di inserimento dei valori

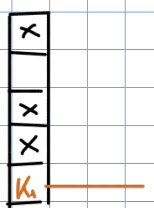
$0 \le i < m$

dobbiamo garantire che
$h(k,i) \neq h(k,J)$ se $i \neq J$

$|T| = 5$

$h(k_1) = <3,2,4,1,0>$ → tutte le possibili posizioni di h

$h(k_2) = <0,2,4,3,1>$



$h(k_1,3)$ → occupata

$h(k_1,2)$ → occupata

$h(k_1,4)$ → posizione valida

Con questo tipo di implementazione le funzioni diventano

```
insert(T, K)
   i ← 0
   while (i < m and T[i] ≠ null) Do
        i ← i + 1
   if (i < m) then   T[i] ← K
```

$O(n)$ $n \le m$

$m = \#$ celle
$n = \#$ elementi

```
Search (T, K)
   i ← 0
   while (i < m and T[i] ≠ null)
        if (T[i] = K) return true
        i ← i + 1
   return false
```

$O(n)$

Quando cancelliamo usiamo un simbolo "D" per indicare che un elemento è stato eliminato, questo cambia le insert: se questa trova "D" può inserire un valore.

Ignoriamo questo comportamento di seguito, abbiamo una tabella hash che non può eliminare

Ricerca senza successo

$x = \#$ ispezioni fatte durante un inserimento

$$E[x] = \sum_{i=1}^{\infty} P_r\{x \geq i\} \qquad h: U \times \{0, 1, \dots m-1\} \to \{0, 1, \dots, m-1\}$$

A_i = troviamo la cella occupata alla i-esima ispezione

$$P_r\{x \geq i\} = P_r\{A_1 \cap A_2 \cap A_3 \cap \dots A_{i-1}\}$$

Probabilità condizionate

$$P_r\{A_1\} \times P_r\{A_2 | A_1\} \times P_r\{A_3 | A_1 \cap A_2\} \times P_r\{A_4 | A_2 \cap A_2 \cap A_3\} \times \dots \times P_r\{A_{i-1} | A_2 \cap \dots \cap A_{i-2}\}$$

$$\frac{n}{m} \geq \frac{n-1}{m-1} \geq \frac{n-2}{m-2} \geq \frac{n-3}{m-3} \geq \frac{n-i+2}{m-1+2}$$

$$\leq \left(\frac{n}{m}\right)^{i-1}$$

$$\leq \sum_{i=1}^{\infty} \left(\frac{n}{m}\right)^{i-1} \to \sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$

serie geometrica

$\alpha = 0{,}5 = \frac{1}{2}$

$$\frac{1}{1-\alpha} = \frac{1}{1-\frac{1}{2}} = 2$$

Ricerca con successo

$$h: U \times \{0, 1, \dots m-1\} \to \{0, 1, \dots, m-1\}$$

$$\alpha_i = \frac{i}{m}$$

$$k_1, k_2, k_3 \dots k_i, k_{i+1} \dots k_m$$

$$k(i+1) \xrightarrow{\text{lavoro per inserimento}} \frac{1}{1-\alpha_i} = \frac{1}{1-\frac{i}{m}} = \frac{1}{\frac{m}{m}-\frac{i}{m}} = \frac{m}{m-i}$$

$$\frac{1}{n} \sum_{i=0}^{n-1} \frac{m}{m-i} = \frac{m}{n} \underbrace{\sum_{i=0}^{n-1} \frac{1}{m-i}}_{} \longrightarrow \frac{m}{n} \sum_{k=m-n+1}^{m}$$

$$\frac{1}{m} + \frac{1}{m-1} + \frac{1}{m-2} + \frac{1}{m-3} + \dots + \frac{1}{m-n+1} = \underbrace{\sum_{k=m-n+1}^{m} \frac{1}{k}}_{}$$

$$\leq \frac{m}{n} \int_{m-n}^{m} \frac{1}{x}\, dx = \frac{m}{n} \left[\ln(m) - \ln(m-n) \right]$$

$$\leq \frac{m}{n} \ln\left(\frac{m}{m-n}\right) = \frac{1}{\alpha} \ln\left(\frac{1}{1-\alpha}\right)$$

$\alpha = 0{,}5 = \frac{1}{2} \to 1{,}4$

$\alpha = 0{,}9 = \frac{9}{10} \Rightarrow 2{,}5$