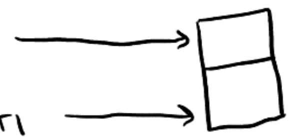
TABELLE HASH

PROBLEMA: RAPPRESENTAZIONE DI INSIEMI DINAMICI CON SUPPORTO EFFICIENTE DELLE OPERAZIONI DI

- INSERIMENTO
- RICERCA (PER CHIAVE)
- CANCELLAZIONE

NOTA: I POSSIBILI ELEMENTI SONO RAPPRESENTATI MEDIANTE RECORD CON DUE CAMPI:

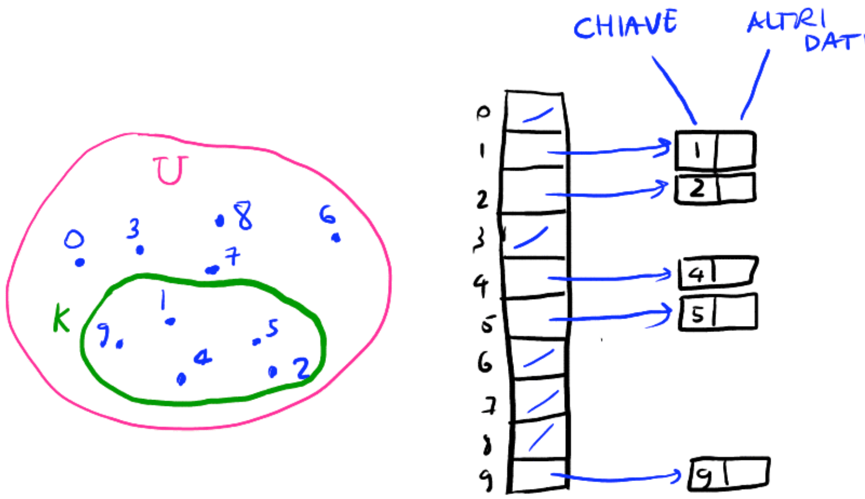
- CHIAVE
- ALTRI DATI



SOLUZIONE MEDIANTE TABELLE AD INDIRIZZAMENTO DIRETTO (ARRAY)

- SIA $U = \{0, 1, 2, ..., m-1\}$ L'UNIVERSO DELLE CHIAVI
- UN INSIEME DINAMICO LE CUI CHIAVI SPAZIANO IN U PUÒ ESSERE RAPPRESENTATO MEDIANTE UN ARRAY DI PUNTATORI $T[0..m-1]$ LE CUI COMPONENTI SONO INIZIALIZZATE A NIL

ESEMPIO



OPERAZIONI

DIRECT-ADDRESS-SEARCH (T, k)

    return $T[k]$

DIRECT-ADDRESS-INSERT (T, x)

    $T[key[x]] := x$

DIRECT-ADDRESS-DELETE (T, x)

    $T[key[x]] := NIL$

COMPLESSITA'

SPAZIO : $O(|U|)$

TEMPO : $O(1)$ PER OPERAZIONE NEL CASO PESSIMO

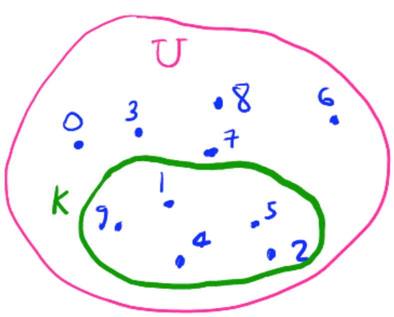
OSSERVAZIONE

- LE TABELLE AD INDIRIZZAMENTO DIRETTO SONO UTILIZZABILI SOLO QUANDO:
  - $|U|$ E' PICCOLO
  - ELEMENTI DISTINTI HANNO CHIAVI DISTINTE

VETTORI DI BIT

- UN CASO SPECIALE SI HA NELLA RAPPRESENTAZIONE DI INSIEMI DI CHIAVI (SENZA DATI SATELLITARI)
- IN TAL CASO SI POSSONO UTILIZZARE ARRAY $T[0..m-1]$ DI BIT

ESEMPIO



0	1	2	3	4	5	6	7	8	9
0	1	1	0	1	1	0	0	0	1

- TALE RAPPRESENTAZIONE SUPPORTA IN MANIERA ABBASTANZA EFFICIENTE ANCHE LE OPERAZIONI DI: - UNIONE, - INTERSEZIONE, - DIFFERENZA INSIEMISTICA

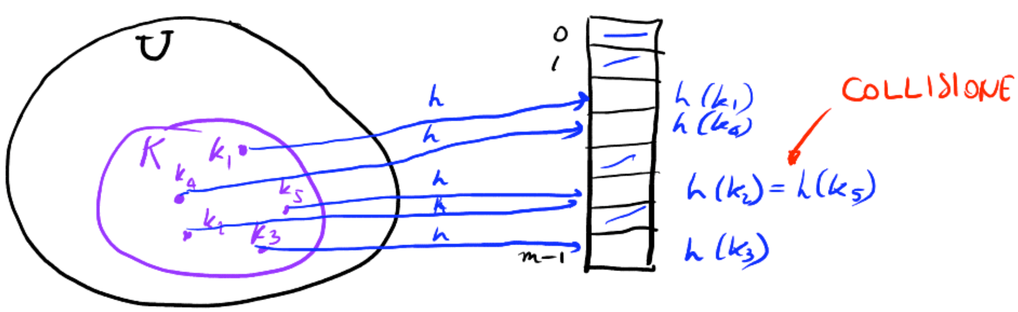
TABELLE HASH

- LE TABELLE HASH RISOLVONO IN PRATICA IL PROBLEMA DELLA RAPPRESENTAZIONE DI INSIEMI DINAMICI QUANDO
  - L'UNIVERSO U DELLE POSSIBILI CHIAVI E' GRANDE (ANCHE INFINITO) E QUINDI RISULTA PROIBITIVO (SE NON IMPOSSIBILE) ALLOCARE UN ARRAY T DI $|U|$ COMPONENTI
  - LA DIMENSIONE DELL'INSIEME DA RAPPRESENTARE E' PICCOLA

- VIENE ALLOCATA UNA "TABELLA HASH" DI DIMENSIONE m CONFRONTABILE CON QUELLA DELL'INSIEME CHE SI INTENDE RAPPRESENTARE
- SI UTILIZZA UNA OPPORTUNA "FUNZIONE HASH"

$$h : U \longrightarrow \{0, 1, \ldots, m-1\}$$

- LA TABELLA HASH VIENE UTILIZZATA COME UNA TABELLA AD INDIRIZZAMENTO DIRETTO, FILTRANDO LE CHIAVI MEDIANTE LA FUNZIONE HASH

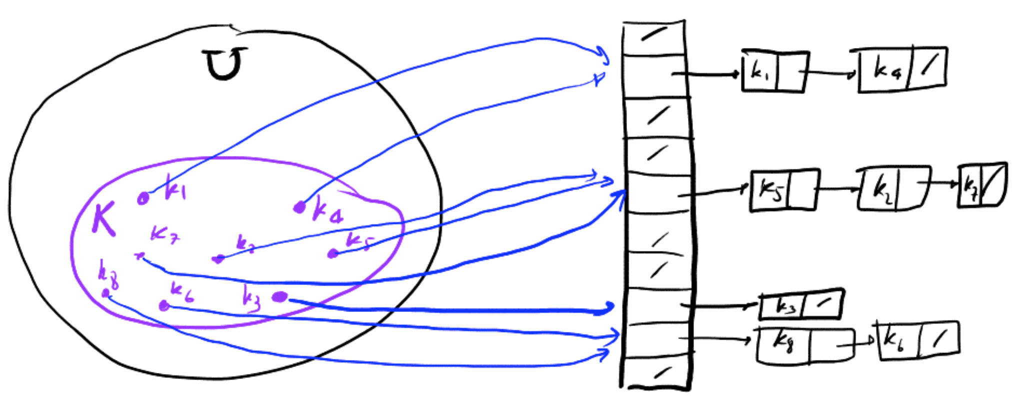


- SE $|U| > |T|$ E' INEVITABILE CHE SI VERIFICHI IL PROBLEMA DELLE COLLISIONI, CIOE' CHE VI SIANO CHIAVI $k_1 \neq k_2$ TALI CHE $h(k_1) = h(k_2)$

- VEDREMO DUE SOLUZIONI AL PROBLEMA DELLE COLLISIONI
  - TABELLE HASH CON CONCATENAZIONE
  - TABELLE HASH AD INDIRIZZAMENTO APERTO

TABELLE HASH CON CONCATENAZIONE

- GLI ELEMENTI CHE COLLIDONO VENGONO INSERITI IN UNA LISTA (CON INSERIMENTO IN TESTA)



OPERAZIONI

CHAINED-HASH-INSERT(T, x)
- si inserisca x in testa alla lista $T[h(key[x])]$

CHAINED-HASH-SEARCH(T, k)
- si cerchi un elemento con chiave k nella lista $T[h(k)]$

CHAINED-HASH-DELETE(T, x)
- si cancelli x dalla lista $T[h(key[x])]$

COMPLESSITA'

- CHAINED-HASH-INSERT(T, x)
  $O(1)$ CASO PESSIMO
- CHAINED-HASH-DELETE(T, x)
  $O(1)$ CASO PESSIMO, PURCHE' LE LISTE SIANO RAPPRESENTATE COME LISTE DOPPIE
- CHAINED-HASH-SEARCH(T, k)
  $O(n)$ CASO PESSIMO,

  $O(1+\alpha)$ CASO MEDIO NELL'IPOTESI DI HASHING UNIFORME SEMPLICE

  (CON n NUMERO DI ELEMENTI, m DIMENSIONE DELLA TABELLA, $\alpha = n/m$ FATTORE DI CARICO)

IPOTESI DI HASHING UNIFORME SEMPLICE

PER OGNI $i \in \{0, 1, \dots, m-1\}$,

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

TEOREMA 1 SOTTO L'IPOTESI DI HASHING UNIFORME SEMPLICE, UNA RICERCA SENZA SUCCESSO RICHIEDE TEMPO $O(1+\alpha)$ IN MEDIA.

DIM. A CAUSA DELL'IPOTESI DI HASHING UNIFORME SEMPLICE, CIASCUNA DELLE m LISTE AVRA' IN MEDIA LUNGHEZZA $\alpha = n/m$ E QUINDI UNA RICERCA SENZA SUCCESSO ESAMINERA' IN MEDIA α ELEMENTI. ■

TEOREMA 2 SOTTO L'IPOTESI DI HASHING UNIFORME SEMPLICE, UNA RICERCA CON SUCCESSO RICHIEDE TEMPO $O(1+\alpha)$ IN MEDIA.

DIM. (INFORMALE)

A CAUSA DELL'IPOTESI DI HASHING UNIFORME SEMPLICE, CIASCUNA DELLE m LISTE AVRA' IN MEDIA LUNGHEZZA $\alpha = n/m$ E QUINDI UNA RICERCA CON SUCCESSO ESAMINERA' IN MEDIA $\alpha/2$ ELEMENTI. ■

INTERPRETAZIONE DELL'ANALISI DI COMPLESSITA'

- SE $n = O(m)$, SI HA $\alpha = \frac{n}{m} = \frac{O(m)}{m} = O(1)$.
- QUINDI SE SI SCEGLIE m PROPORZIONALE AD n, LA RICERCA CON O SENZA SUCCESSO RICHIEDE IN MEDIA TEMPO $O(1)$.

FUNZIONI HASH

- UNA "BUONA" FUNZIONE HASH DEVE SODDISFARE APPROSSIMATIVAMENTE L'IPOTESI DI HASHING UNIFORME SEMPLICE
- PERO', PER VERIFICARE TALE IPOTESI E' NECESSARIO CONOSCERE
  - LA DISTRIBUZIONE DI PROBABILITA' CON LA QUALE VENGONO SELEZIONATI GLI ELEMENTI DALL'UNIVERSO U
  - E' INOLTRE IMPORTANTE CHE TALI ELEMENTI SIANO SELEZIONATI IN MANIERA INDIPENDENTE L'UNO DALL'ALTRO

ESEMPIO

SE $U = \{ k : 0 \leq k < 1 \}$ E GLI ELEMENTI SONO SELEZIONATI DA U SECONDO UNA DISTRIBUZIONE UNIFORME, ALLORA LA FUNZIONE $h : U \longrightarrow \{0, 1, \dots, m-1\}$ DEFINITA DA:

$$h(k) = \lfloor k \cdot m \rfloor$$

SODDISFA L'IPOTESI DI HASHING UNIFORME SEMPLICE.

INTERPRETAZIONE DELLE CHIAVI COME NUMERI NATURALI

- NELLA MAGGIOR PARTE DELLE FUNZIONI HASH, VIENE ASSUNTO CHE $U = \mathbb{N} = \{0, 1, 2, \dots\}$
- QUINDI, PER UTILIZZARE TALI FUNZIONI HASH OCCORRERA' MAPPARE U IN $\mathbb{N}$

ESEMPIO

U = INSIEME DELLE STRINGHE FINITE DI CARATTERI ASCII A 7 BIT

$$pt \longrightarrow (112, 116) \longrightarrow 112 \cdot 128 + 116 = 14452$$

FUNZIONI HASH CON IL METODO DELLA DIVISIONE

- SIA m LA DIMENSIONE DELLA TABELLA HASH.
  SI PONE: $h(k) = k \bmod m$

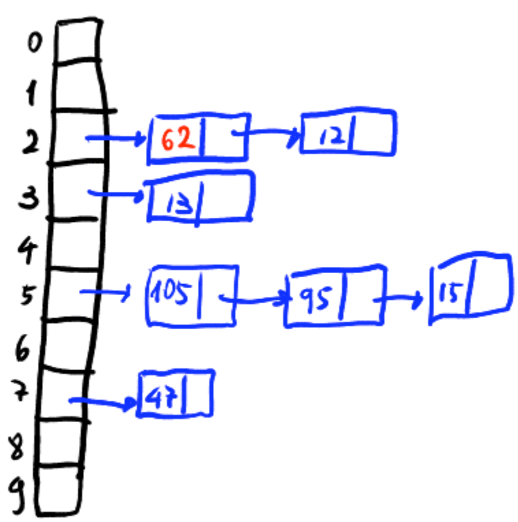
- TALE METODO E' MOLTO EFFICIENTE, MA OCCORRE AVERE CURA DI SCEGLIERE UN VALORE m CHE APPROSSIMI BENE L'IPOTESI DI HASHING UNIFORME SEMPLICE

ES. - SE $m = 2^p$, $h(k)$ DIPENDE DAI p BIT DI ORDINE INFERIORE

- UNA BUONA SCELTA CONSISTE IN GENERE NEL SELEZIONARE PER m UN NUMERO PRIMO ABBASTANZA DISCOSTO DA POTENZE DI 2

ES. $n = 2000 \longrightarrow m = 701 \longrightarrow \alpha \simeq 3$

$m = 10$, 47, 12, 15, 95, 62, 13, 105



METODO DELLA DIVISIONE

$h(x) = x \bmod 10$

$h(47) = 7$
$h(12) = 2$
$h(15) = 5$
$h(95) = 5$
$h(62) = 2$
$h(13) = 3$
$h(105) = 5$

FUNZIONI HASH CON IL METODO MOLTIPLICATIVO

- SIA $0 < A < 1$ UNA COSTANTE FISSATA.
  SI PONE $h(k) = \lfloor m\,(kA \bmod 1) \rfloor$
  (DOVE $kA \bmod 1 = kA - \lfloor kA \rfloor$)
- LA SCELTA DI m NON E' CRITICA
- CONVIENE UTILIZZARE IL VALORE
  $A = (\sqrt{5} - 1)/2 \simeq 0{,}6180339887\ldots$

TABELLE HASH AD INDIRIZZAMENTO APERTO

- NELLE TABELLE HASH CON CONCATENAZIONE PARTE DELLA MEMORIA E' IMPEGNATA CON PUNTATORI
- SI PUÒ EVITARE DI UTILIZZARE PUNTATORI MANTENENDO TUTTI I DATI ALL'INTERNO DELLA STESSA TABELLA, UTILIZZANDO LO SCHEMA DELL' INDIRIZZAMENTO APERTO
- IN TAL CASO SI AVRA' $\alpha \leq 1$
- L'INSERIMENTO DI UN NUOVO ELEMENTO UTILIZZERÀ UNA SEQUENZA DI SCANSIONE DIPENDENTE DAL VALORE DELLA CHIAVE k
- LA MEDESIMA SEQUENZA DI SCANSIONE SARA' UTILIZZATA NELLA RICERCA

- PER GENERARE LE SEQUENZE DI SCANSIONE SARANNO UTILIZZATE FUNZIONI HASH DEL TIPO:

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

(m E' LA DIMENSIONE DELLA TABELLA)

- DATA LA CHIAVE k, SARA' UTILIZZATA LA SEGUENTE SEQUENZA:

$$\langle h(k,0), h(k,1), \dots, h(k,m-1) \rangle$$

- PER AUMENTARE L'EFFICIENZA DELLE TABELLE HASH AD INDIRIZZAMENTO APERTO E' IMPORTANTE CHE LE SEQUENZE DI SCANSIONE SIANO PERMUTAZIONI DI $\langle 0, 1, \dots, m-1 \rangle$

```
HASH-INSERT (T, k)
   i := 0
   repeat  j := h(k, i)
           if  T[j] = NIL  then
                  T[j] := k
                  return j
           else
                  i := i + 1
   until  i = m
   error ("hash table overflow")
```

```
HASH-SEARCH (T, k)
  i := 0
  repeat  j := h(k, i)
       if  T[j] = k  then
            return  j
       i := i+1
  until  T[j] = NIL  or  i = m
  return  NIL
```

NOTA: TALE SCHEMA **NON** SUPPORTA L'OPERAZIONE DI CANCELLAZIONE

COMPLESSITA'

- I SEGUENTI RISULTATI DI COMPLESSITA' PRESUPPONGONO L'IPOTESI DI HASHING UNIFORME:

PER OGNI PERMUTAZIONE π DI $\langle 0, 1, \dots, m-1 \rangle$ VALE

$$Pr\{\langle h(k,0), h(k,1), \dots, h(k, m-1)\rangle = \pi \mid k \in U\} = \frac{1}{m!}$$

(CIOE' TUTTE LE PERMUTAZIONI SONO EQUIPROBABILI)

TEOREMA IL NUMERO MEDIO DI SCANSIONI IN UNA RICERCA SENZA SUCCESSO IN UNA TABELLA HASH AD INDIRIZZAMENTO APERTO E' AL PIÙ $\frac{1}{1-\alpha}$ (PER $\alpha = \frac{n}{m} < 1$), ASSUMENDO L'IPOTESI DI HASHING UNIFORME ∎

OSSERVAZIONE

- SE α E' COSTANTE, IL TEMPO RICHIESTO E' $O(1)$
- IN PARTICOLARE, SE $\alpha = 0.5$ (CIOE' LA TABELLA HASH E' PIENA PER METÀ), OCCORRERANNO AL PIÙ $\frac{1}{1-0.5} = 2$ SCANSIONI,

  SE $\alpha = 0.9$ (CIOE' LA TABELLA E' PIENA AL 90%) OCCORRERANNO AL PIÙ $\frac{1}{1-0.9} = 10$ SCANSIONI

COROLLARIO NELLE IPOTESI DEL TEOREMA PRECEDENTE, L'INSERIMENTO DI UN NUOVO ELEMENTO IN UNA TABELLA HASH CON FATTORE DI CARICO $\alpha < 1$ RICHIEDE AL PIÙ $\frac{1}{1-\alpha}$ SCANSIONI ∎

TEOREMA IL NUMERO MEDIO DI SCANSIONI IN UNA RICERCA CON SUCCESSO IN UNA TABELLA HASH AD INDIRIZZAMENTO APERTO E' AL PIÙ $\frac{1}{\alpha} \ln \frac{1}{1-\alpha}$ (PER $\alpha = \frac{n}{m} < 1$), ASSUMENDO L'IPOTESI DI HASHING UNIFORME E SUPPONENDO CHE TUTTE LE CHIAVI PRESENTI NELLA TABELLA HANNO LA STESSA PROBABILITÀ DI ESSERE CERCATE ∎

ESEMPIO

- SE $\alpha = 0.5$, IL NUMERO MEDIO DI SCANSIONI E' $\leq 1{,}387$.
- SE $\alpha = 0.9$, IL NUMERO MEDIO DI SCANSIONI E' $\leq 2{,}559$.

RICERCA CON INSUCCESSO IN IPOTESI DI HASHING UNIFORME

p_i = probabilità che vengano scandite "esattamente" i locazioni occupate

$$\#\text{ medio locazioni scandite} = 1 + \sum_{i=0}^{\infty} i\, p_i$$

q_i = probabilità che vengano scandite "almeno" i locazioni occupate

$$q_1 = p_1 + p_2 + p_3 + \dots$$
$$q_2 = \quad p_2 + p_3 + \dots$$
$$q_3 = \qquad p_3 + p_4 + \dots$$

$$q_1 = p_1 + p_2 + p_3 + \dots$$
$$q_2 = \quad p_2 + p_3 + \dots$$
$$q_3 = \qquad p_3 + p_4 + \dots$$

$$\vdots \quad \vdots \qquad \vdots$$

$$\sum_{i=1}^{\infty} q_i = p_1 + 2\cdot p_2 + 3\cdot p_3 + 4\cdot p_4 + \dots = \sum_{i=1}^{\infty} i \cdot p_i = \sum_{i=0}^{\infty} i \cdot p_i$$

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

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

$$q_1 = \alpha = \frac{n}{m}$$

$$q_2 = \frac{n}{m} \cdot \frac{n-1}{m-1} \leq \left(\frac{n}{m}\right)^2$$

$$\vdots$$

$$q_j = \frac{n}{m} \cdot \frac{n-1}{m-1} \cdot \dots \cdot \frac{n-j+1}{m-j+1} \leq \left(\frac{n}{m}\right)^j$$

$$\#\text{ medio} \leq \frac{1}{1-\alpha}$$

RICERCA CON SUCCESSO IN IPOTESI DI HASHING UNIFORME

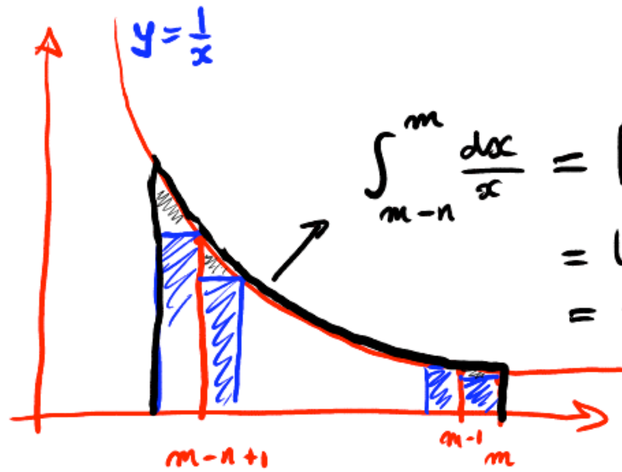
$\frac{1}{\alpha} \ln \frac{1}{1-\alpha}$

$k_1, k_2, k_3, \ldots, k_n$ NELL'ORDINE IN CUI SONO STATE INSERITE

$k_i \longrightarrow$ # MEDIO DI SCANSIONI $\leq \dfrac{1}{1-\frac{i-1}{m}}$

$$\#\text{MEDIO} \leq \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1-\frac{i-1}{m}} = \frac{1}{n} \sum_{i=1}^{n} \frac{m}{m-i+1} = \frac{1}{\alpha} \sum_{i=1}^{n} \frac{1}{m-i+1}$$

$$\sum_{i=1}^{n} \frac{1}{m-i+1} = \frac{1}{m} + \frac{1}{m-1} + \ldots + \frac{1}{m-n+1} \leq \ln \frac{1}{1-\alpha}$$

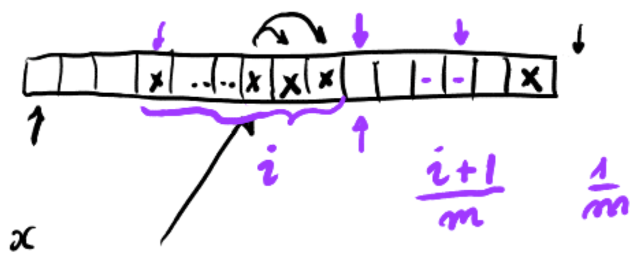


$$\int_{m-n}^{m} \frac{dx}{x} = \left[\ln x\right]_{m-n}^{m}$$

$$= \ln m - \ln (m-n)$$

$$= \ln \frac{m}{m-n} =$$

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



m sequenze di scansione $\ll m!$

$c = 1$

0 $\quad h'(x)$ – HASH AUSILIARIA

1 $\quad h'(x) + 1$

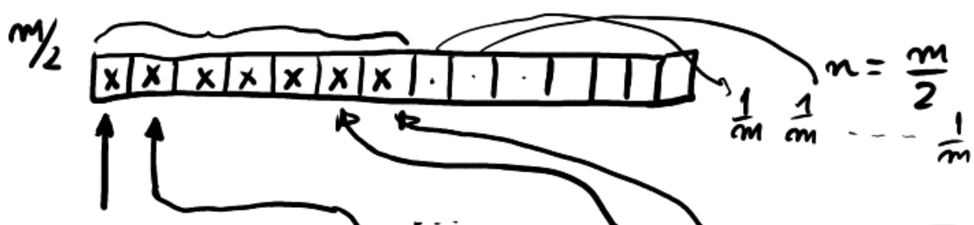
2 $\quad h'(x) + 2$

$\vdots$

$(h'(x) + c\,i) \bmod m$

$(c, m) = 1$

AGGLOMERAZIONE PRIMARIA

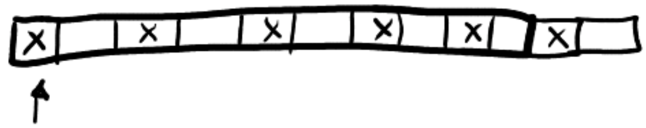


$$\frac{1}{m}\left(\frac{m}{2}+1\right) + \frac{1}{m}\cdot\frac{m}{2} + \dots + \frac{1}{m}3 + \frac{1}{m}2 + \frac{1}{m}\cdot\frac{m}{2}$$

$$= \frac{1}{m}\sum_{i=1}^{m/2+1} i + \frac{1}{m}\left(\frac{m}{2}-1\right)$$

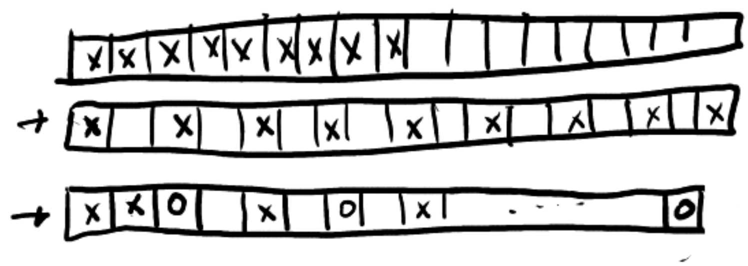
$$= \frac{1}{m}\,\frac{(m/2+1)(m/2+2)}{2} + \frac{1}{2} - \frac{1}{m} = \frac{1}{m}\,\frac{\frac{m^2}{4} + \frac{3}{2}m + 2}{2} + \frac{1}{2} - \frac{1}{m}$$

$$= \frac{m}{8} + \frac{3}{4} + \cancel{\frac{1}{m}} + \frac{1}{2} - \cancel{\frac{1}{m}} = \frac{m}{8} + \frac{5}{4} \simeq \frac{n}{4}$$



2 1 2 1 2 1 2 1 2 1 2 1 =

$$\frac{1}{m}\left(2\cdot\frac{m}{2} + 1\cdot\frac{m}{2}\right) = \frac{1}{m}\cdot\frac{3}{2}\cdot m = \frac{3}{2}$$



CALCOLO DELLE SEQUENZE DI SCANSIONE

METODO DELLA SCANSIONE LINEARE

- SIA $h' : U \longrightarrow \{0, 1, \ldots, m-1\}$ UNA FUNZIONE HASH AUSILIARIA
- PONIAMO $h(k,i) = (h'(k) + i) \bmod m$
- TALE SCHEMA SOFFRE PERO' DEL PROBLEMA DELL'AGGLOMERAZIONE PRIMARIA, CONSISTENTE NELLA FORMAZIONE DI LUNGHE SEQUENZE DI SLOT CONTIGUI OCCUPATI
- INFATTI, LA PROBABILITA' CHE UNO SLOT PRECEDUTO DA i SLOT GIA' OCCUPATI VENGA OCCUPATO E' $\frac{(i+1)}{m}$.

METODO DELLA SCANSIONE QUADRATICA

- SIA $h' : U \to \{0, 1, \dots, m-1\}$ UNA FUNZIONE HASH AUSILIARIA
- SIANO $c_1, c_2 \in \mathbb{Q}$ DUE COSTANTI, CON $c_2 \neq 0$
- PONIAMO:

$$h(k,i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$$

- TALE SCHEMA SOFFRE DEL PROBLEMA MENO GRAVE DELL' AGGLOMERAZIONE SECONDARIA, IN QUANTO SE $k \neq k'$ SONO TALI CHE $h'(k) = h'(k')$, ALLORA

$$\langle h(k,0), \dots, h(k,m-1) \rangle = \langle h(k',0), \dots, h(k',m-1) \rangle$$

- PER AUMENTARE L'EFFICIENZA DEL METODO DELLA SCANSIONE QUADRATICA, E' IMPORTANTE CHE LE COSTANTI c_1 E c_2 SIANO SCELTE IN MODO DA GARANTIRE CHE TUTTE LE SEQUENZE DI SCANSIONE PRODOTTE SIANO PERMUTAZIONI DI $\langle 0, 1, \dots, m-1 \rangle$

- SI VERIFICA CHE LA SCELTA

$$c_1 = c_2 = \frac{1}{2}, \quad m = 2^p$$

SODDISFA LA CONDIZIONE DI SOPRA

HASHING QUADRATICO

$$h(x,i) = \left(h'(x) + c_1 i + c_2 i^2\right) \bmod m$$

$$\boxed{\begin{array}{l} m = 2^r \\ c_1 = c_2 = \frac{1}{2} \end{array}}$$

$0 \le i, j \le 2^r - 1 \qquad i \ne j$

$h(x,i) = h(x,j)$

$$\cancel{h'(x)} + \frac{1}{2} i + \frac{1}{2} i^2 \equiv \cancel{h'(x)} + \frac{1}{2} j + \frac{1}{2} j^2 \pmod{2^r}$$

$$\frac{1}{2}(i-j) + \frac{1}{2}(i+j)(i-j) \equiv 0 \pmod{2^r}$$

$$\frac{1}{2}(i-j)(i+j+1) \equiv 0 \pmod{2^r}$$

$$2^r \mid \frac{1}{2}(i-j)(i+j+1)$$

$$2^{r+1} \mid (i-j)(i+j+1)$$

$2^{r+1} \mid i-j \rightarrow i = j$

ASSURDO

$\cancel{2^{r+1} \mid i+j+1}$

$0 < i+j+1 < 2^{r+1}$

$$2 \mid i-j \Longleftrightarrow 2 \nmid i+j+1$$

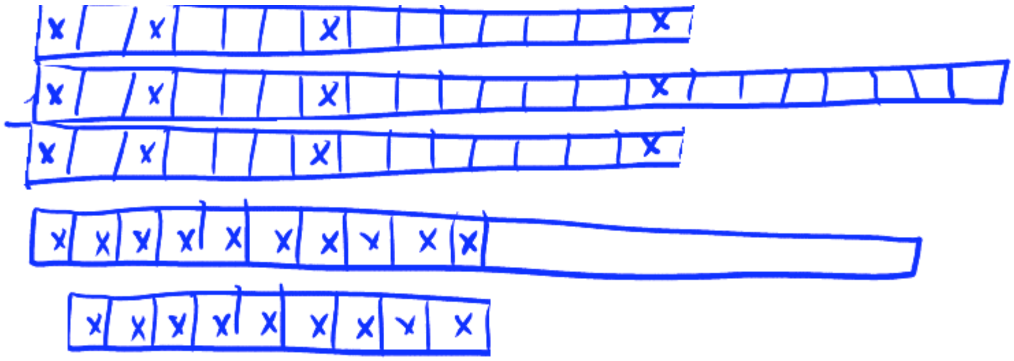
$$\Updownarrow$$

$$2 \mid i-j+2j = i+j$$

$0 \le |i-j| \le i+j \le 2^{r+1}-2$

$i, j \le 2^r - 1$

$$2^{r+1} \mid i-j \rightarrow 2^{r+1} \mid |i-j| \quad \boxed{i+j+1 \le 2^{r+1} - 1}$$



AGGLOMERAZIONE SECONDARIA

$$h'(x) = h'(y)$$

METODO DELL'HASHING DOPPIO

- SIANO $h_1, h_2 : U \longrightarrow \{0,1,\dots,m-1\}$ DUE FUNZIONI HASH AUSILIARIE
- PONIAMO:

$$h(k,i) = (h_1(k) + h_2(k)\cdot i) \bmod m$$

- PERCHE' LE SEQUENZE DI SCANSIONE SIANO PERMUTAZIONI DI $\langle 0,1,\dots,m-1\rangle$ E' NECESSARIO E SUFFICIENTE CHE $h_2(k)$ SIA PRIMO CON m, PER OGNI $k \in U$.
- UNA POSSIBILE SCELTA E':

$$m = 2^p$$
$$h_2 : U \longrightarrow \{1, 3, 5, \dots, m-1\}$$

- UN'ALTRA SCELTA POSSIBILE E':

m PRIMO

$h_2 : U \longrightarrow \{1, 2, \ldots, m-1\}$

ESEMPIO

m PRIMO

$h_1(k) = k \bmod m$

$h_2(k) = 1 + (k \bmod m')$ (CON $m' < m$)

$m = 701$

$m' = 700$

$k = 123456$

$h_1(k) = 123456 \bmod 701 = 80$

$h_2(k) = 1 + (123456 \bmod 700) = 257$

INDIRIZZAMENTO APERTO E CANCELLAZIONE

- PER NON INTERROMPERE LE SEQUENZE DI SCANSIONE E' SUFFICIENTE MARCARE DELETED GLI SLOT DA CUI SI CANCELLANO ELEMENTI
- L'ANALISI DI COMPLESSITA' DOVRA' TENERE CONTO ANCHE DEGLI SLOT MARCATI DELETED NEL COMPUTO DEL FATTORE DI CARICO α

```
HASH-INSERT'(T, k)
   i := 0
   repeat  j := h(k, i)
           if  T[j] = NIL  OR  T[j] = DELETED  then
                 T[j] := k
                 return j
           else
                 i := i + 1
   until  i = m
   error ("hash table overflow")
```