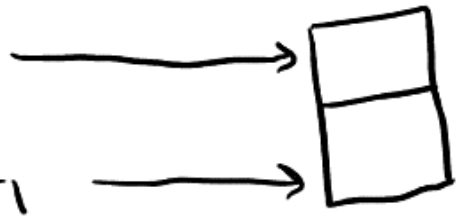
# TABELLE HASH

**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 O PIU' CAMPI:

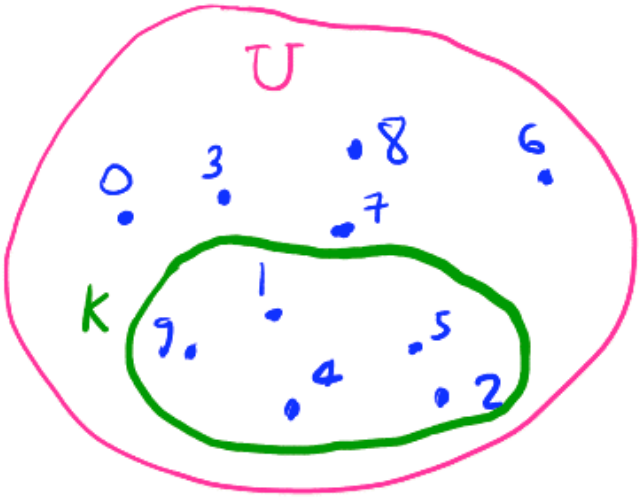
- CHIAVE
- ALTRI DATI

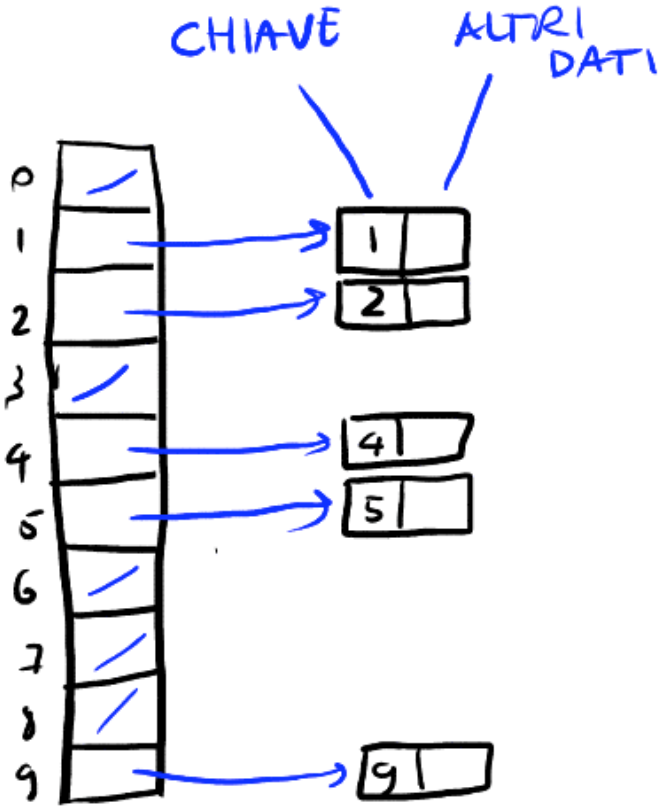


# SOLUZIONE MEDIANTE TABELLE AD INDIRIZZAMENTO DIRETTO (ARRAY)

- SIA $U = \{0, 1, 2, \ldots, M-1\}$ 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

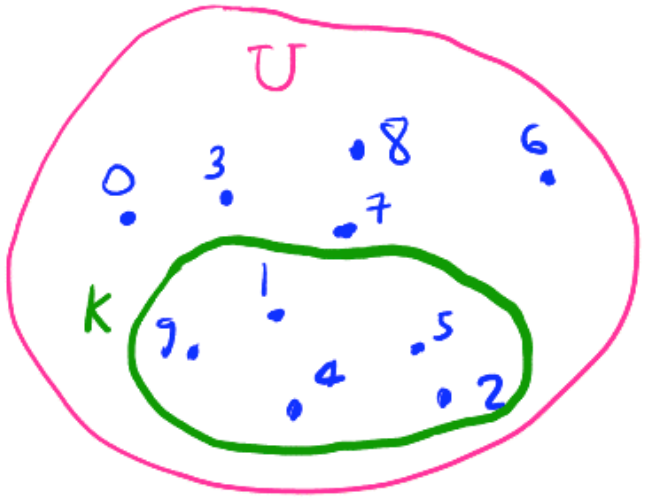
## OSSERVAZIONE

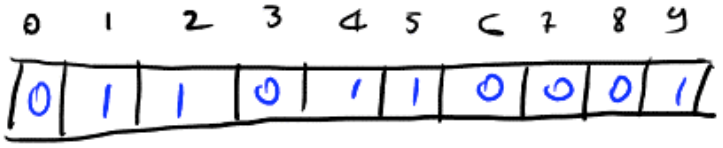
- LE TABELLE AD INDIRIZZAMENTO DIRETTO SONO UTILIZZABILI SOLO QUANDO:
  - $|U|$ E' PICCOLO
  - ELEMENTI DISTINTI HANNO CHIAVI DISTINTE

# VETTORI DI BIT

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

## ESEMPIO




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

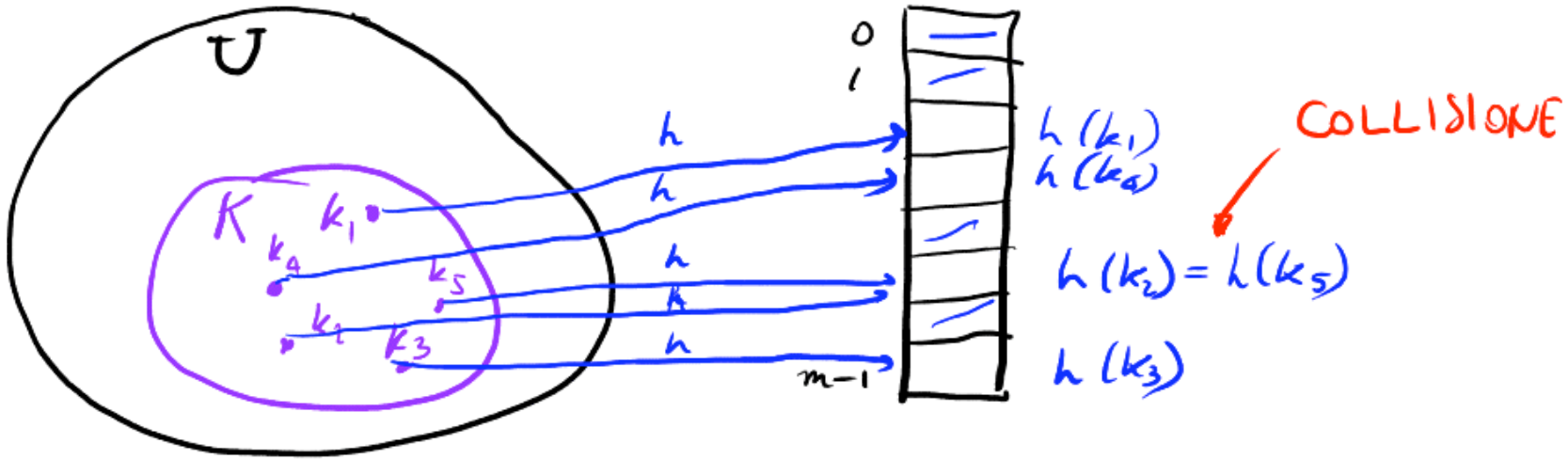
# TABELLE HASH

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

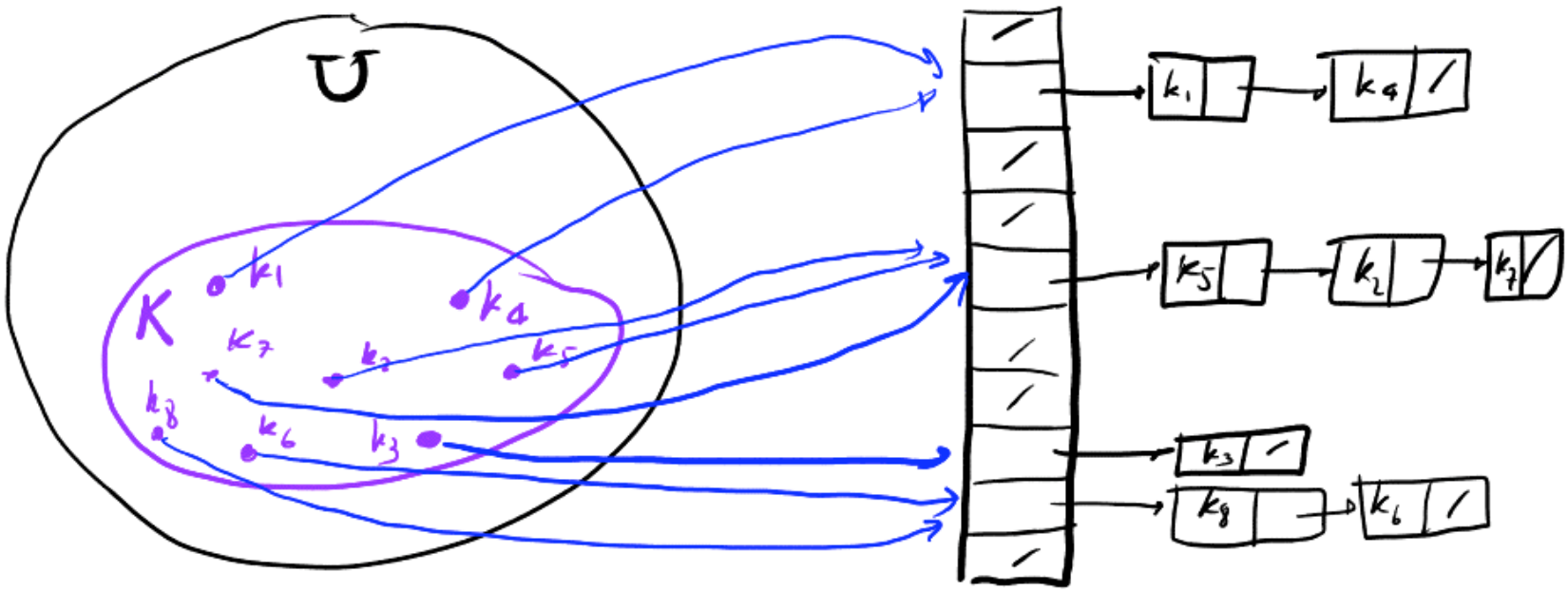


- SE $|U| > |T|$ E' INEVITABILE CHE SI POSSA VERIFICARE IL PROBLEMA DELLE COLLISIONI, CIOE' CHE VI SIANO CHIAVI $k_1 \neq k_2$ TALI CHE $h(k_1) = h(k_2)$

- 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 $T[h(key[x])]$

## COMPLESSITÀ

- CHAINED-HASH-INSERT $(T, x)$

  $O(1)$ CASO PESSIMO

- CHAINED-HASH-DELETE $(T, x)$

  $O(1)$ CASO PESSIMO, PURCHÈ LE LISTE SIANO RAPPRESENTATE COME LISTE DOPPIE

- CHAINED-HASH-SEARCH $(T, k)$

  $\Theta(n)$ CASO PESSIMO,

  $\Theta(1+\alpha)$ 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 $i \in \{0, 1, \ldots, m-1\}$,

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

**TEOREMA 1** NELL'IPOTESI DI HASHING UNIFORME SEMPLICE, UNA RICERCA SENZA SUCCESSO RICHIEDE TEMPO $\Theta(1+\alpha)$ 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 $\alpha$ ELEMENTI. ■

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

**DIM.** - SIANO $k_1, k_2, \dots, k_n$ LE CHIAVI NELL'ORDINE IN CUI SONO STATE INSERITE IN TABELLA.

- SUPPONIAMO CHE TUTTE ABBIANO LA STESSA PROBABILITA' DI ESSERE RICERCATE
- LA RICERCA DELLA CHIAVE $k_i$ ANALIZZERA' IN MEDIA $1 + \frac{n-i}{m}$ RECORD
- QUINDI IN MEDIA UNA RICERCA CON SUCCESSO ANALIZZERA' $\frac{1}{n} \sum_{i=1}^{n} \left(1 + \frac{n-i}{m}\right)$ RECORD

÷

SI HA:

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

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

$$= 1+\frac{n}{m}-\frac{1}{n\cdot m}\,\frac{n(n+1)}{2}$$

$$= 1+\frac{n}{m}-\frac{n}{2m}-\frac{1}{2m}$$

$$= 1+\frac{n}{2m}-\frac{1}{2n}\cdot\frac{n}{m}$$

$$= 1+\frac{\alpha}{2}-\frac{\alpha}{2n} \Rightarrow \bigcirc(1+\alpha)$$

■

# INTERPRETAZIONE DELL'ANALISI DI COMPLESSITÀ

- 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 **SAREBBE** 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

$h : U \longrightarrow \{0, 1, \ldots, m-1\}$ 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 $U = \mathbb{N} = \{0, 1, 2, \ldots\}$
- QUINDI, PER UTILIZZARE TALI FUNZIONI HASH OCCORRERA' MAPPARE $U$ IN $\mathbb{N}$ QUALORA, $U \not\subseteq \mathbb{N}$

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

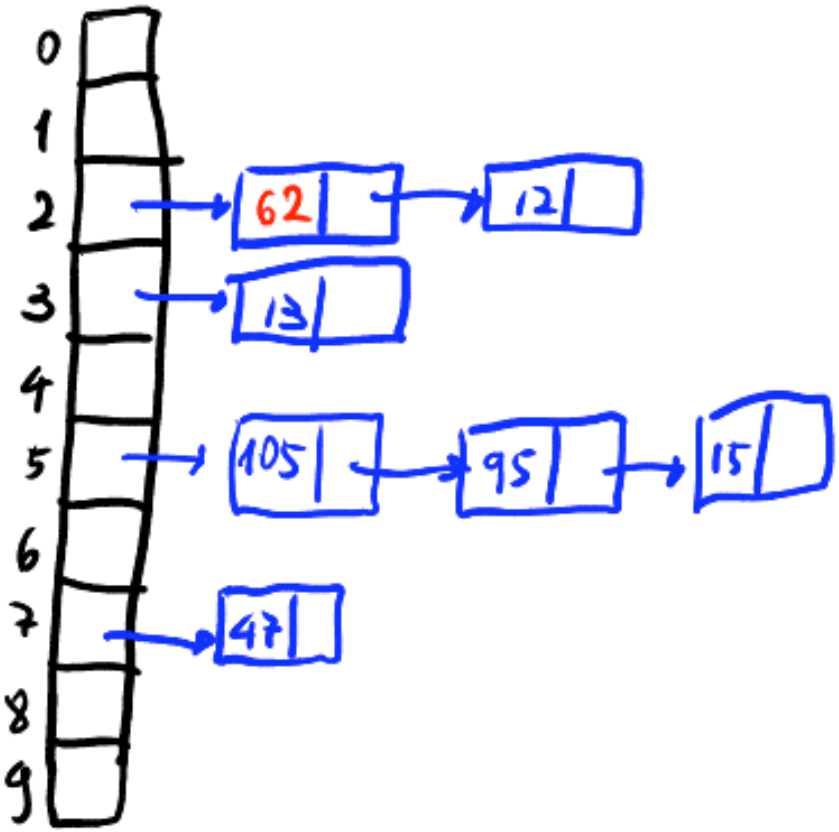
- TALE METODO È MOLTO EFFICIENTE, MA OCCORRE AVERE CURA DI SCEGLIERE UN VALORE $m$ CHE APPROSSIMI BENE L'IPOTESI DI HASHING UNIFORME SEMPLICE

  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

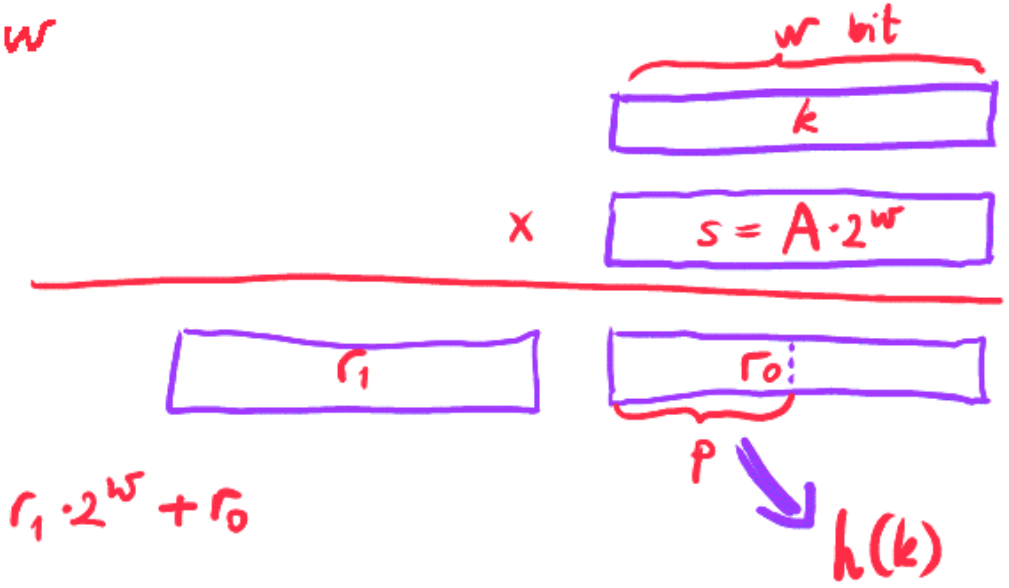
$h(x) = x \bmod 10$

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

# FUNZIONI HASH CON IL METODO DELLA MOLTIPLICAZIONE

- 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
  $A = (\sqrt{5} - 1)/2 \simeq 0.6180339887 \ldots$

- TIPICAMENTE SI SCEGLIE $m = 2^p$
- CIO' RENDE IL CALCOLO DI $h(k)$ PARTICOLARMENTE EFFICIENTE IN SITUAZIONI IN CUI, POSTO $w$ LA DIMENSIONE DI UNA PAROLA, SI HA
  - $0 \leq k < 2^w$
  - $A = \frac{s}{2^w}$, CON $0 < s < 2^w$
  - $p < w$




$k \cdot s = r_1 \cdot 2^w + r_0$

ESEMPIO:

$k = 123456$

$p = 14$

$m = 2^{14} = 16384$

$w = 32$

$A = \frac{s}{2^{32}} = \frac{2654435769}{2^{32}} \simeq \frac{\sqrt{5}-1}{2}$

QUINDI:

$k \cdot s = 327706022297664 = (76300 \cdot 2^{32}) + 17612864$

$r_1 = 76300$

$r_0 = 17612864$

I 14 BIT PIÙ SIGNIFICATIVI DI $r_0$ FORNISCONO IL VALORE

$h(k) = 67$

# HASHING UNIVERSALE

- AL FINE DI EVITARE CHE POSSANO ESISTERE INSIEMI DI CHIAVI CHE CAUSINO SEMPRE UN ALTO NUMERO DI COLLISIONI, E' STATO PROPOSTO LO SCHEMA DELL' HASHING UNIVERSALE, CHE PREVEDE CHE LA FUNZIONE HASH SIA SELEZIONATA IN MANIERA RANDOM DA UNA CERTA FAMIGLIA DI FUNZIONI HASH, CHE GODE DI OPPORTUNE PROPRIETA',

- SIA $\mathcal{H}$ UNA FAMIGLIA DI FUNZIONI HASH $h: U \to \{0..m-1\}$

**DEFINIZIONE** $\mathcal{H}$ SI DICE UNIVERSALE SE

$$(\forall x,y \in U)\left(x \neq y \longrightarrow |\{h \in \mathcal{H} : h(x) = h(y)\}| = \frac{|\mathcal{H}|}{m}\right)$$ ■

**PROPRIETÀ** SIA $\mathcal{H}$ UNIVERSALE E SIANO $x, y \in U$ TALI CHE $x \neq y$, ALLORA

$$\Pr\{h \in \mathcal{H} : h(x) = h(y)\} = \frac{|\{h \in \mathcal{H} : h(x) = h(y)\}|}{|\mathcal{H}|} = \frac{1}{m}.$$

(CIOÈ, LA PROBABILITÀ DI AVERE UNA COLLISIONE SU DUE ELEMENTI $x, y$ SELEZIONANDO $h$ DA $\mathcal{H}$ È UGUALE ALLA PROBABILITÀ DI OTTENERE UNA COLLISIONE SELEZIONANDO IN MANIERA RANDOM I DUE VALORI HASH SU $x$ E $y$) ■

**TEOREMA** SIA $\mathcal{H}$ UNA FAMIGLIA UNIVERSALE DI FUNZIONI HASH $h: U \to \{0, \dots, m-1\}$ E SIA $K \subseteq U$ UN INSIEME DI CHIAVI TALE CHE $|K| = n \leq m$ (DA INSERIRE IN UNA TABELLA HASH CON CONCATENAZIONE, DI DIMENSIONE $m$).
SIA $x \in K$ UNA CHIAVE FISSATA.
ALLORA IL NUMERO MEDIO DI COLLISIONI CON $x$ OTTENUTE SELEZIONANDO IN MANIERA RANDOM UNA FUNZIONE $h$ IN $\mathcal{H}$ E' MINORE DI 1.

**DIM.**

- PER OGNI $h \in \mathcal{H}$, IL NUMERO DI COLLISIONI CON $x$ GENERATE DA $h$ E':

$$|\{y \in K \setminus \{x\} : h(y) = h(x)\}|$$

- QUINDI, IL NUMERO MEDIO DI COLLISIONI CON $x$ E':

$$\frac{1}{|\mathcal{H}|}\sum_{h\in\mathcal{H}}|\{y\in K\setminus\{x\} : h(y)=h(x)\}|$$

$$=\frac{1}{|\mathcal{H}|}\sum_{h\in\mathcal{H}}\sum_{\substack{y\in K\setminus\{x\}\\ h(y)=h(x)}}1=\frac{1}{|\mathcal{H}|}\sum_{y\in K\setminus\{x\}}\sum_{\substack{h\in\mathcal{H}\\ h(y)=h(x)}}1$$

$$=\frac{1}{|\mathcal{H}|}\sum_{y\in K\setminus\{x\}}|\{h\in\mathcal{H} : h(y)=h(x)\}|$$

$$=\frac{1}{|\mathcal{H}|}\sum_{y\in K\setminus\{x\}}\frac{|\mathcal{H}|}{m}=\frac{|K\setminus\{x\}|}{m}=\frac{n-1}{m}<\frac{n}{m}\leq 1$$

■

# COSTRUZIONE DI UNA FAMIGLIA UNIVERSALE DI FUNZIONI HASH

- SIA $U = \{0, 1, 2, \dots, M-1\}$ (E QUINDI $M = |U|$)
- SIA $m$ UN NUMERO PRIMO
- SIA $r = \lceil \log_m M \rceil$, DA CUI $M \leq m^r$
- QUINDI, SE $x \in U$ SI HA: $x < M \leq m^r$, DA CUI:

$$x \leq m^r - 1 = (m-1)(1 + m + \dots + m^{r-1})$$
$$= (m-1)\cdot m^{r-1} + (m-1)\cdot m^{r-2} + \dots + (m-1)\cdot m + (m-1)$$

CIOÈ $x$ È ESPRIMIBILE IN BASE $m$ CON AL PIÙ $r$ CIFRE: $x_{r-1}\ x_{r-2} \dots x_1\ x_0$

$$x = x_{r-1}\cdot m^{r-1} + x_{r-2}\cdot m^{r-2} + \dots + x_1 \cdot m + x_0$$

- SI OTTIENE QUINDI UN'APPLICAZIONE DA $U$ IN $\{0..m-1\}^r$

$$x \longmapsto (x_0, x_1, \ldots, x_{r-1})$$

- PER OGNI $a \in \{0..m-1\}^r$, CON $a = (a_0, a_1, \ldots, a_{r-1})$, DEFINIAMO $h_a : U \to \{0..m-1\}$ PONENDO:

$$h_a(x) = \left( \sum_{i=0}^{r-1} a_i x_i \right) \bmod m$$

OVE $x = (x_0, x_1, \ldots, x_{r-1})$

- PONIAMO $\mathcal{H} = \{ h_a : a \in \{0..m-1\}^r \}$

- DIMOSTREREMO CHE $\mathcal{H}$ E' UNA FAMIGLIA UNIVERSALE DI FUNZIONI HASH

**LEMMA 1** $\quad |\mathcal{H}| = m^r$

**DIM** E' SUFFICIENTE VERIFICARE CHE L'APPLICAZIONE

$a \mapsto h_a$ E' INIETTIVA.

- SIANO $a, b \in \{0..m-1\}^r$ TALI CHE $a \neq b$ E
- SUPPONIAMO CHE $a_0 \neq b_0$.
- SI CONSIDERI $x = (1, 0, 0, \dots, 0)$ (CIOE' $x_0 = 1$, $x_2 = \dots = x_{r-1} = 0$).
- SI HA:

$$h_a(x) = \left(\sum_{i=0}^{r-1} a_i x_i\right) \bmod m = a_0 \neq b_0 = \left(\sum_{i=0}^{r-1} b_i x_i\right) \bmod m = h_b(x)$$

CIOE' $h_a \neq h_b$, DA CUI L'INIETTIVITA' DI $a \mapsto h_a$

E QUINDI $|\mathcal{H}| = m^r$. ∎

**LEMMA 2** SIANO $x, y \in U$ TALI CHE $x \neq y$.

ALLORA: $|\{h_a \in \mathcal{H} : h_a(x) = h_a(y)\}| = m^{r-1}$,

DA CUI $|\{h_a \in \mathcal{H} : h_a(x) = h_a(y)\}| = \frac{|\mathcal{H}|}{m}$

(CIOE' $\mathcal{H}$ E' UNA FAMIGLIA UNIVERSALE DI FUNZIONI HASH)

**DIM.** SIANO $x = (x_0, x_1, \dots, x_{r-1})$ E $y = (y_0, y_1, \dots, y_{r-1})$ E

SUPPONIAMO CHE $x_0 \neq y_0$.

- E' SUFFICIENTE ESIBIRE UNA CORRISPONDENZA BIUNIVOCA $\sigma$ TRA $\mathcal{H}_{x,y} = \{h_a \in \mathcal{H} : h_a(x) = h_a(y)\}$ E $\{0, \dots, m-1\}^{r-1}$.
- DATA $h_a \in \mathcal{H}_{x,y}$, PONIAMO $\sigma(h_a) = (a_1, \dots, a_{r-1})$
- VERIFICHIAMO CHE $\sigma$ E' UNA CORRISPONDENZA BIUNIVOCA

## INIETTIVITÀ

- SIANO $h_a, h_b \in \mathcal{H}_{x,y}$ TALI CHE $h_a \neq h_b$.
- ALLORA $a \neq b$.
- POICHÉ $h_a(x) = h_a(y)$, SI HA:

$$\sum_{i=0}^{r-1} a_i x_i \equiv \sum_{i=0}^{r-1} a_i y_i \pmod{m}$$

$$a_0(x_0 - y_0) \equiv \sum_{i=1}^{r-1} a_i (y_i - x_i) \pmod{m}$$

- ANALOGAMENTE

$$b_0(x_0 - y_0) \equiv \sum_{i=1}^{r-1} b_i (y_i - x_i) \pmod{m}$$

È QUINDI

$$(a_0 - b_0)(x_0 - y_0) \equiv \sum_{i=1}^{r-1} (a_i - b_i)(y_i - x_i) \pmod{m}$$

- PERTANTO, SE $\sigma(h_a) = \sigma(h_b)$, CIOE' SE $(a_1 \dots a_{r-1}) = (b_1 \dots b_{r-1})$,
  SI AVREBBE: $(a_0 - b_0)(x_0 - y_0) \equiv 0 \pmod{m}$
  DA CUI $a_0 = b_0$ (ESSENDO PER IPOTESI $x_0 \neq y_0$)
- MA ALLORA SI AVREBBE $(a_0 \dots a_{r-1}) = (b_0 \dots b_{r-1})$,
  CIOE' $a = b$, ASSURDO.

## SURIETTIVITA'

- SIA $(a_1 .. a_{r-1}) \in \{0,..,m-1\}^{r-1}$ E SI CONSIDERI L'EQUAZIONE IN $\alpha$:

$$\alpha(x_0 - y_0) \equiv \sum_{i=1}^{r-1} a_i (y_i - x_i) \pmod{m}$$

- POICHE' $x_0 \neq y_0$ ED $m$ E' PRIMO, TALE EQUAZIONE AMMETTE UN'UNICA SOLUZIONE, ESPRIMIBILE COME

$$\bar{\alpha} = \sum_{i=1}^{r-1} a_i (y_i - x_i)(x_0 - y_0)^{-1} \pmod{m},$$

- MA ALLORA, POSTO $\bar{a} = (\bar{\alpha}, a_1, \dots, a_{r-1})$, SI HA:

$h_{\bar{a}} \in \mathcal{H}_{x,y}$ E $\sigma(h_{\bar{a}}) = (a_1 .. a_{r-1})$, DA CUI LA SURIETTIVITA'.

■

# 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 AVRÀ $\alpha \leq 1$
- L'INSERIMENTO DI UN NUOVO ELEMENTO UTILIZZERÀ UNA SEQUENZA DI SCANSIONE DIPENDENTE DAL VALORE DELLA CHIAVE $k$
- LA MEDESIMA SEQUENZA DI SCANSIONE SARÀ UTILIZZATA NELLA RICERCA

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

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

($m$ È LA DIMENSIONE DELLA TABELLA)

- DATA LA CHIAVE $k$, SARÀ UTILIZZATA LA SEGUENTE SEQUENZA:

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

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

```
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 $\pi$ DI $\langle 0, 1, \ldots, m-1 \rangle$ VALE

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

(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 $\alpha$ 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 PROBABILITA' 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

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

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

$q_1 = P_1 + P_2 + P_3 + \cdots$

$q_2 = P_2 + P_3 + \cdots$

$q_3 = P_3 + P_4 + \cdots$

$$q_1 = p_1 + p_2 + p_3 + \cdots$$
$$q_2 = \quad p_2 + p_3 + \cdots$$
$$q_3 = \quad\quad p_3 + p_4 + \cdots$$
$$\vdots \quad\quad\quad \vdots$$

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

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

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

$$\vdots$$

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

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

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

$\#\,medio \le \frac{1}{1-\alpha}$

# RICERCA CON SUCCESSO IN IPOTESI DI HASHING UNIFORME

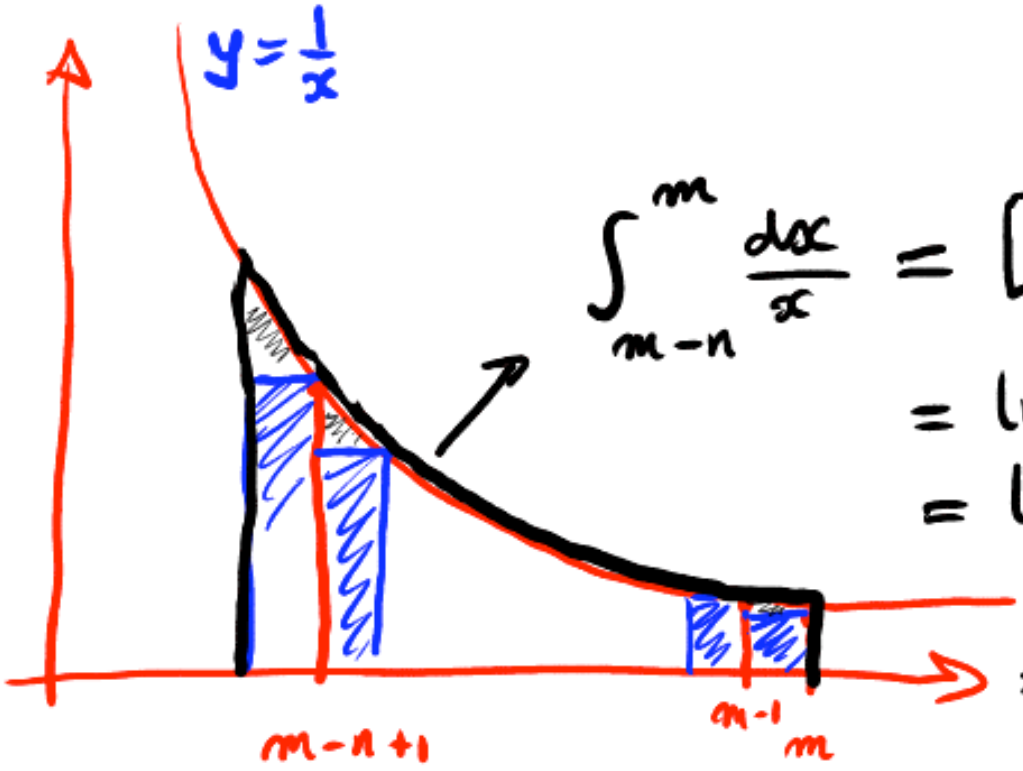
$\frac{1}{\alpha} \ln \frac{1}{1-\alpha}$

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

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

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

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




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

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

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

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

# 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' $\frac{(i+1)}{m}$.

$i$ $\quad \frac{i+1}{m} \quad \frac{1}{m}$

$x$

$m$ sequenze di scansione $<<m!$

$c$ $\quad =1$

0 $\quad h'(x)$ $\quad$ – $\quad$ HASH AUSILIARIA

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

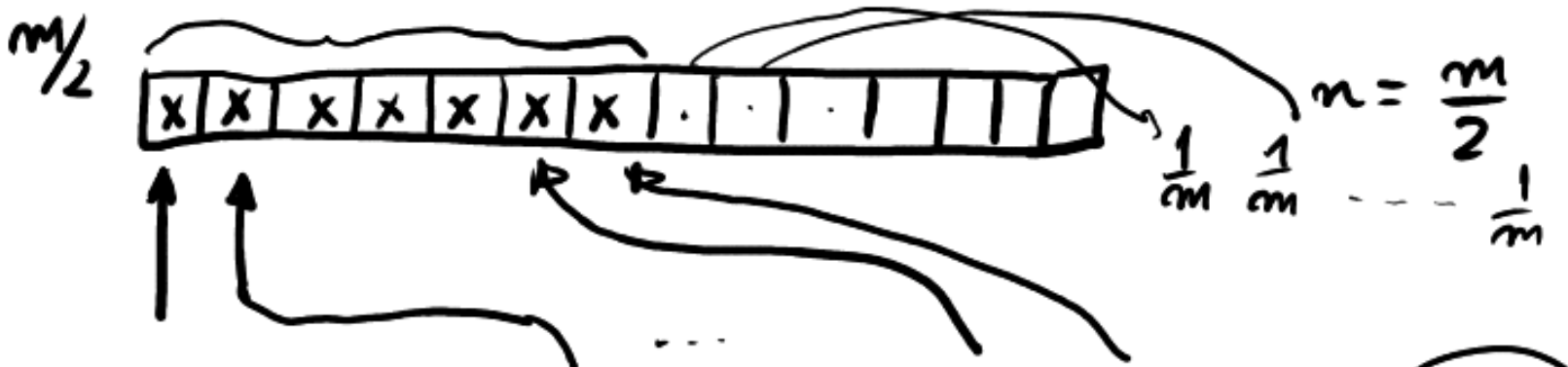
2 $\quad h'(x)+2$

$\vdots$

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

$(c, m) = 1$

AGGLOMERAZIONE PRIMARIA

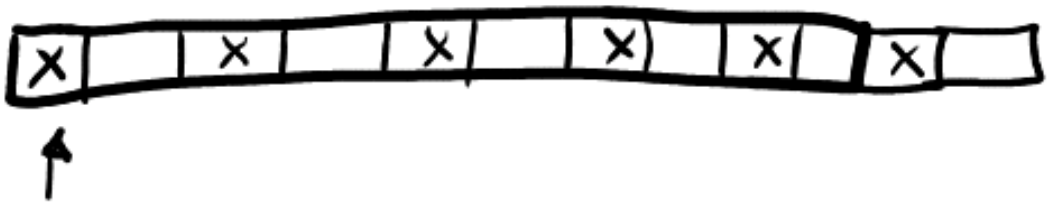



$$\frac{1}{m}\left(\frac{m}{2}+1\right)+\frac{1}{m}\cdot\frac{m}{2}+\dots+\frac{1}{m}3+\frac{1}{m}2+\underbrace{\frac{1}{m}\cdot\frac{m}{2}}$$

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

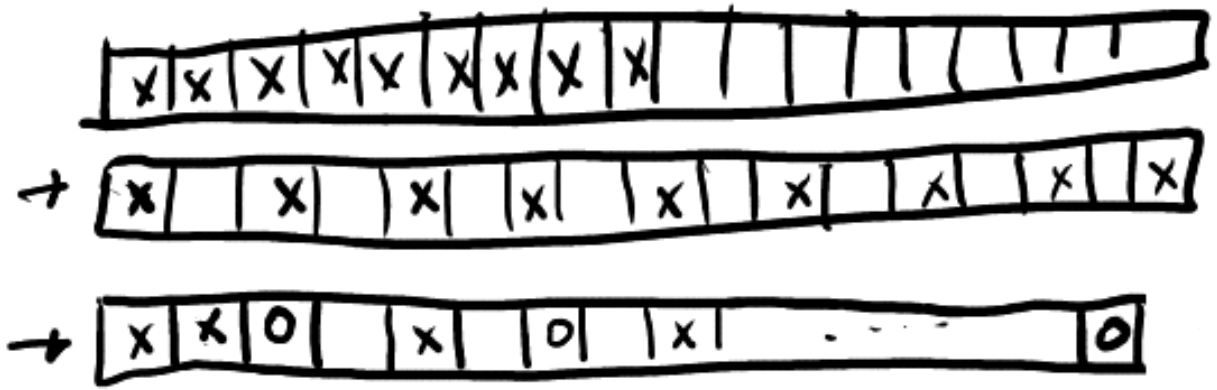
$$=\frac{1}{m}\frac{(m/2+1)(m/2+2)}{2}+\frac{1}{2}-\frac{1}{m}=\frac{1}{m}\frac{\frac{m^2}{4}+\frac{3}{2}m+2}{2}+\frac{1}{2}-\frac{1}{m}$$

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



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

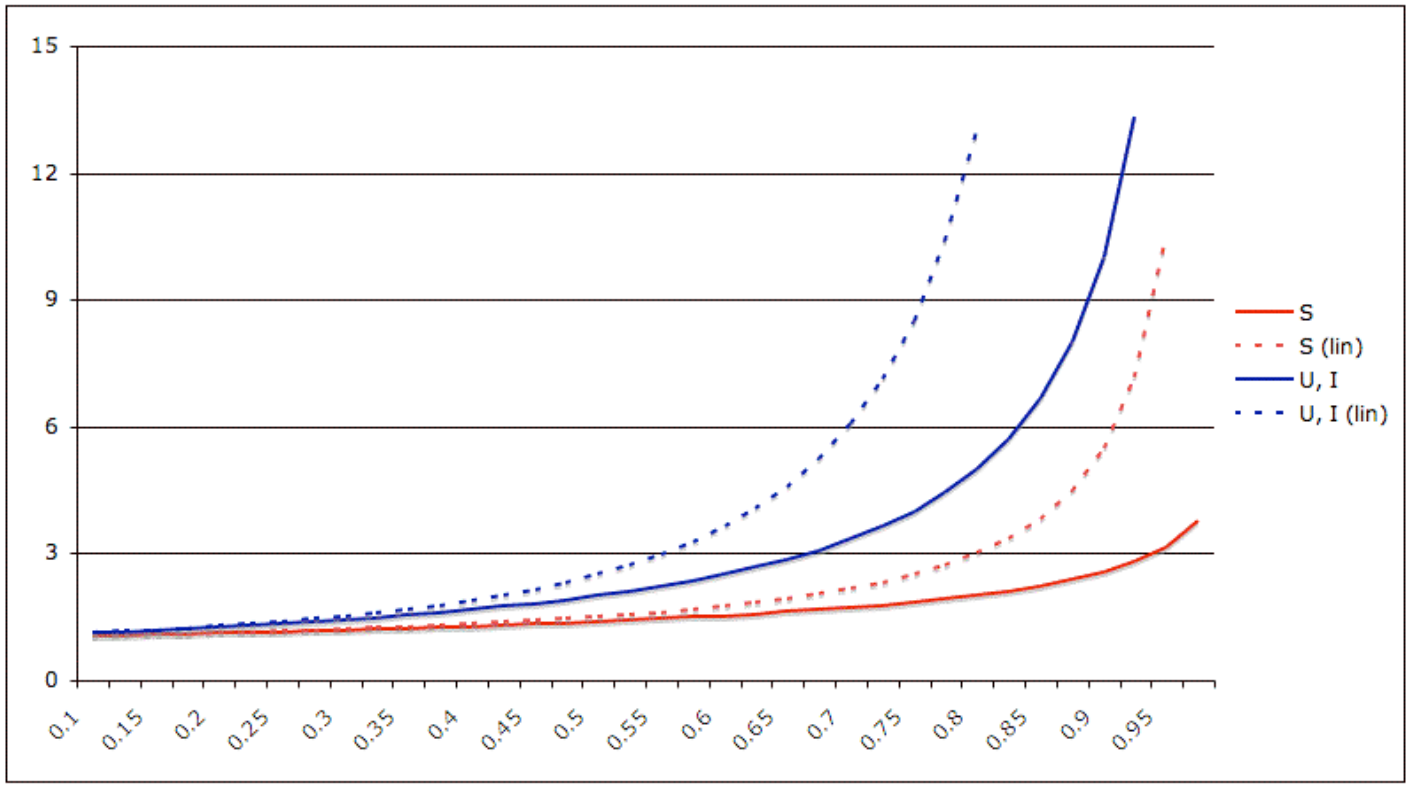
$$\frac{1}{m}\left(2 \cdot \frac{m}{2} + 1 \cdot \frac{m}{2}\right) = \frac{1}{m} \cdot \frac{3}{2} \cdot m = \frac{3}{2}$$



$c = 1$

$c = 2$

$m - 1$




RICERCA SENZA SUCCESSO (SCANSIONE LINEARE) $\frac{1}{2}\left(1+\frac{1}{(1-\alpha)^2}\right)$

RICERCA CON SUCCESSO (SCANSIONE LINEARE) $\frac{1}{2}\left(1+\frac{1}{1-\alpha}\right)$

# METODO DELLA SCANSIONE QUADRATICA

- SIA $h' : U \to \{0, 1, \ldots, m-1\}$ 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

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

- 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 $\langle 0, 1, \ldots, M-1 \rangle$

- SI VERIFICA CHE LA SCELTA

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

SODDISFA LA CONDIZIONE DI SOPRA

# HASHING QUADRATICO

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

$m = 2^r$

$c_1 = c_2 = \frac{1}{2}$

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

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

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

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

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

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

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

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

ASSURDO

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

$2 \mid i-j \iff 2 \mid i+j+1$

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

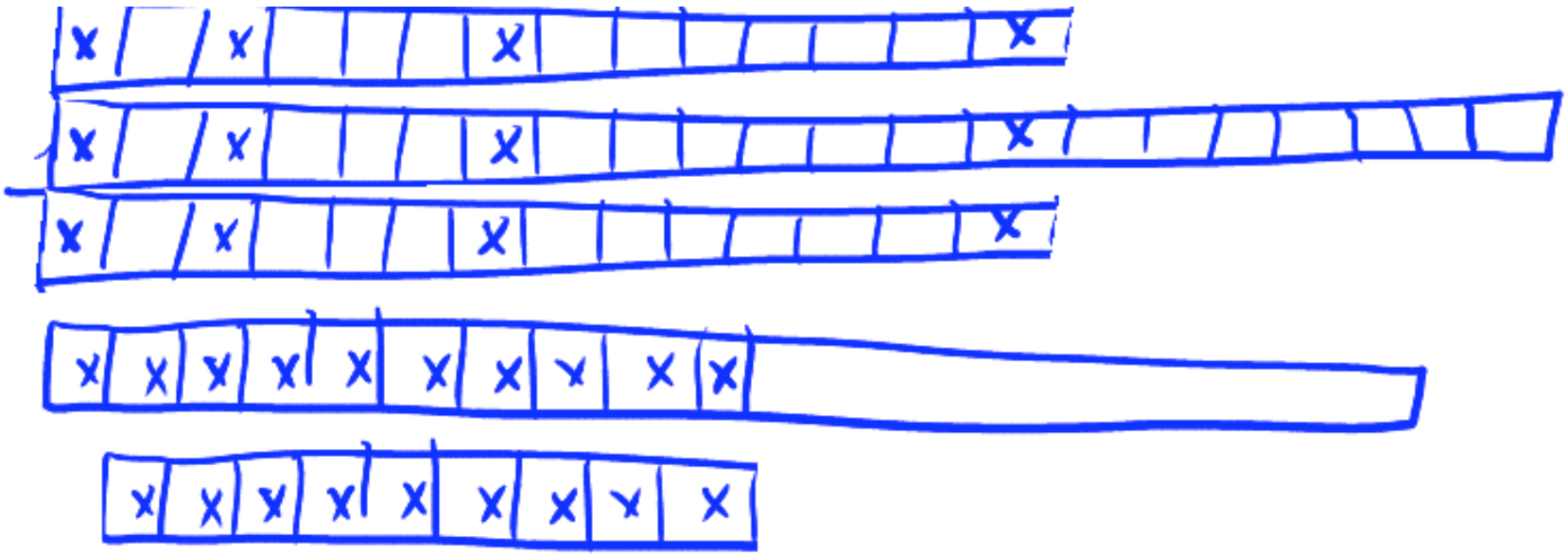
$0 \leq |i-j| \leq i+j \leq 2^{r+1}-2$

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

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

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

$i+j+1 \leq 2^{r+1}-1$



AGGLOMERAZIONE SECONDARIA

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

## METODO DELL'HASHING DOPPIO

- SIANO $h_1, h_2 : U \longrightarrow \{0, 1, \ldots, m-1\}$ 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 $\langle 0, 1, \ldots, m-1 \rangle$ E' NECESSARIO E SUFFICIENTE CHE $h_2(k)$ SIA PRIMO CON $m$, PER OGNI $k \in U$.

- UNA POSSIBILE SCELTA E':

$m = 2^p$

$h_2 : U \longrightarrow \{1, 3, 5, \ldots, m-1\}$

- 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 $\alpha$

```
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")
```