

# Indici Hash

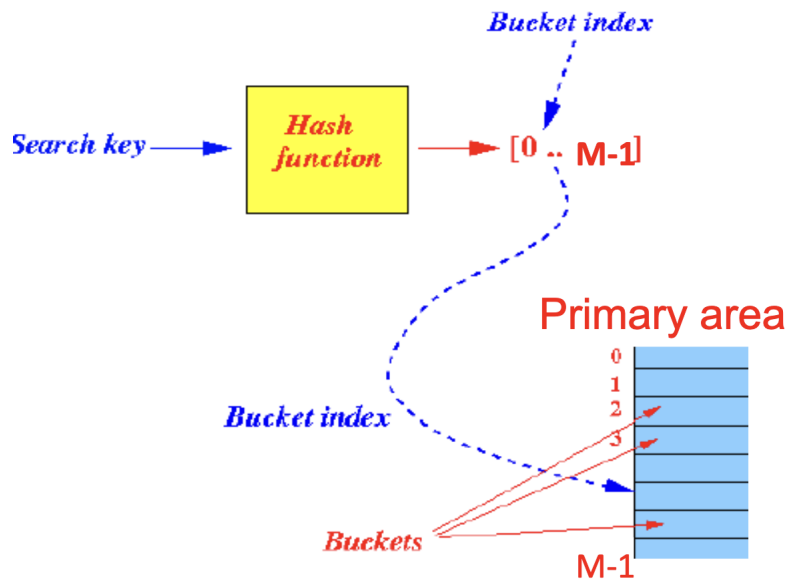
Lorenzo Vaccarecci

5 Aprile 2024

L'uso di indici ad albero ha lo svantaggio di richiedere la scansione di una struttura dati, memorizzata su disco, per localizzare i dati. Questo perché le associazioni  $(k_i, r_i)$  vengono mantenute in forma esplicita, come record in un file. Gli indici hash al contrario mantengono le associazioni  $(k_i, r_i)$  in modo implicito, tramite l'uso di una funzione hash, definita sul dominio della chiave di ricerca.

## 1 Caratteristiche Generali

- Una funzione hash su  $K$  è una funzione **surgettiva**  $H(D_K) \rightarrow \{0, \dots, M-1\}$
- $M$  costante



*A ogni valore della funzione hash corrisponde un indirizzo in area primaria.*

## Esempio

$I_A(R) \quad H(D_A \rightarrow \{0, \dots, 2\})$ , la funzione  $H$  restituisce il numero del bucket.

- $H(1) = 1$
- $H(2) = 2$
- $H(3) = 0$
- $H(5) = 2$
- $H(10) = 1$

1. creo bucket index: 

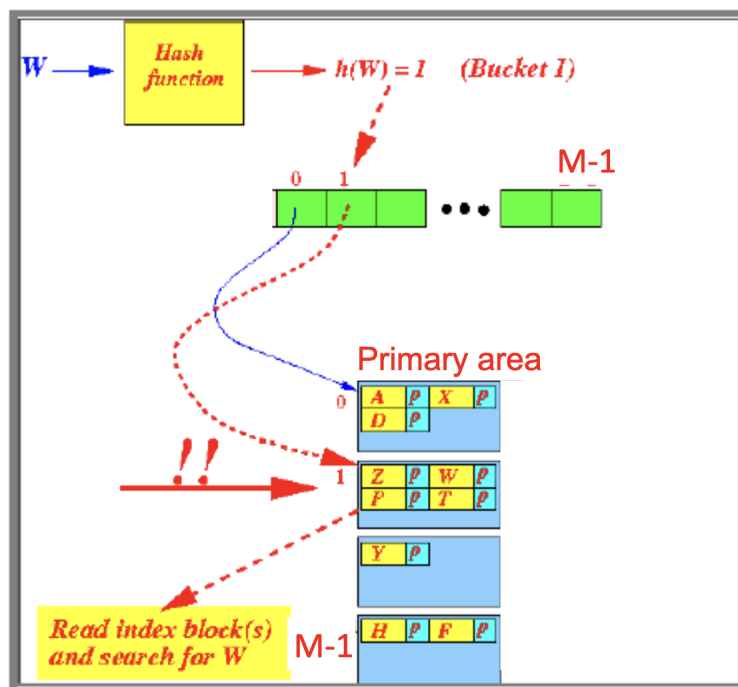
0	1	2
---	---	---

A	B
1	a
2	a
3	b
5	c
10	d
3	f

$\Rightarrow$

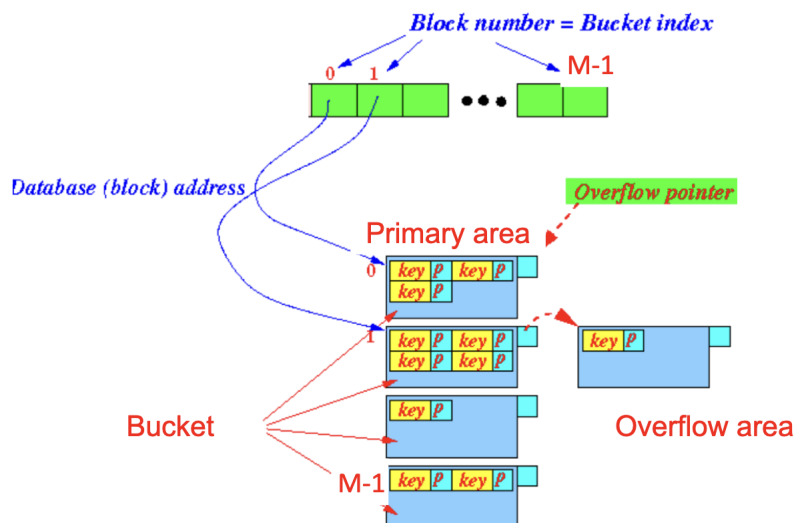
Bucket 0	
3	b
3	f
Bucket 1	
1	a
10	d
Bucket 2	
2	a
5	c

## 2 Ricerca per uguaglianza



*Non supporta la ricerca per intervallo in quanto dovrei conoscere tutti i valori dell'intervallo in quanto la funzione hash non mantiene l'ordine.*

### 3 Inserimento e Trabocchi



Se il bucket è pieno, si alloca un nuovo blocco (*trabocco*) dove c'è spazio chiamato **area di overflow** e se anche l'area di overflow è piena se ne crea un'altra e così via.

#### 3.1 Ricerca

