Tabelle Hash: scansione lineare

Si supponga di operare su di una tabella hash ad indirizzamento aperto contenente 10 posizioni (m=10) e inizialmente vuota. La funzione hash ad essa associata è basata sul metodo della scansione lineare e utilizza una funzione hash ausiliaria basata sul metodo della divisione. Si supponga infine di aver inserito, nell'ordine dato, le chiavi 56, 13, 4, 16, 30 e 79. Indicare la probabilità che la successiva chiave venga inserita nella cella i-esima, per i compreso tra 0 e 9. N.B. Si supponga che la funzione hash ausiliaria goda della proprietà dell'hashing uniforme semplice.

m = 10; h(k) = k mod m = k mod 10

h(56) = 56 mod 10 = 6 [__, __, __, __, __, __, 56, __, __, __]

h(13) = 13 mod 10 = 3 [__, __, __, 13, __, __, 56, __, __, __]

h(4) = 4 mod 10 = 4 [__, __, __, 13, 4, __, 56, __, __, __]

h(16) = 16 mod 10 = 6 [__, __, __, 13, 4, __, 56, 16, __, __]

h(30) = 30 mod 10 = 0 [30, __, __, 13, 4, __, 56, 16, __, __]

h(79) = 79 mod 10 = 9 [30, __, __, 13, 4, __, 56, 16, __, 79]

Probabilità a fine inserimento
[30, __, __, 13, 4, __, 56, 16, __, 79]

P(0) = 0%
P(1) = 3/10 = 30%
P(2) = 1/10 = 10%
P(3) = 0%
P(4) = 0%
P(5) = 3/10 = 30%
P(6) = 0%
P(7) = 0%
P(8) = 3/10 = 30%
P(9) = 0%

Si consideri una tabella hash ad indirizzamento aperto formata da 10 celle e la cui configurazione, dopo i primi 5 inserimenti è la seguente:

$$[X - XX - -X - -X] \tag{1}$$

In tale contesto si supponga che il simbolo X rappresenti una cella occupata e che il simbolo - rappresenti una cella libera. Si forniscano le probabilità di ciascuna cella di essere scelta come destinataria dalla successiva operazione di inserimento.

Soluzione

L'indirizzamento aperto è una gestione alternativa a le liste per gestire quelle che sono le collisioni.
Idea: memorizzare tutte le chiavi nella tabella stessa Ogni slot contiene una chiave oppure nil

- Inserimento: Se lo slot prescelto è utilizzato, si cerca uno slot "alternativo"
- Ricerca: Si cerca nello slot prescelto, e poi negli slot "alternativi" fino a quando

Ispezione: Un'ispezione è l'esame di uno slot durante la ricerca, e la funzione hash viene estesa nel seguente modo

$$H : \mathcal{U} \times \overbrace{[0 \ldots m-1]}^{\text{Numero ispezione}} \to \overbrace{[0 \ldots m-1]}^{\text{Indice vettore}}$$

sequenza di ispezione Una sequenza di ispezione [H(k, 0), H(k, 1), . . . , H(k, m 1)] è una permutazione degli indici [0, . . . , m 1] corrispondente all'ordine in cui vengono esaminati gli slot.

- Non vogliamo esaminare ogni slot più di una volta
- Potrebbe essere necessario esaminare tutti gli slot nella tabella

hashing uniforme

La situazione ideale prende il nome di hashing uniforme, in cui ogni chiave ha la stessa probabilità di avere come sequenza di ispezione una qualsiasi delle m! permutazioni di [0, . . . , m 1].

- Generalizzazione dell'hashing uniforme semplice

- Nella realtà:
  - È difficile implementare il vero hashing uniforme
  - Ci si accontenta di ottenere almeno una permutazione
- Tecniche diffuse:
  - Ispezione lineare
  - Ispezione quadratica
  - Doppio hashing

nell'esercizio utilizzando un ispezione lineare:

$$H(k,i) = (H1(k) + hi) mod m \tag{2}$$

le probabilità risultati sono le seguenti: [X, _, X, X, _, _, X, _, _, X]

P(0) = 0%
P(1) = 3/10 = 30%
P(2) = 0%
P(3) = 0%
P(4) = 3/10 = 30%
P(5) = 1/10 = 10%
P(6) = 0%
P(7) = 2/10 = 20%
P(8) = 1/10 = 10%
P(9) = 0%

utlizzando strettamente il concetto di indirizzamento aperto: Se lo slot prescelto è utilizzato, si cerca uno slot "alternativo", dobbiamo considerare che lo slot scelto ha la probabilità di essere scelto come primo, quindi 1/10 + la probabilità che ogni slot scelto prima era occupato e quindi siamo arrivati a questo slot libero, dal momento che gli slot occupati sono 5 la probabilità sarà per ogni chiave di 6/10 cioè del 60%