Hash con Metodo delle Divisioni vs Hash con Metodo delle Moltiplicazioni

Marco De Groskovskaja

March 13, 2023

1 Introduzione

Il Metodo delle Divisioni (HDM) e il Metodo delle Moltiplicazioni (HMM) sono due comuni funzioni di hashing per calcolare l' indice della tabella in cui inserire il valore dato. Questa relazione esplora le differenze di complessità tra i due metodi e valuta l'efficienza delle tabelle che li implementano per le operazioni di inserimento, ricerca e cancellazione, non che per il numero di collisioni che occorrono.

2 Hashing con Indirizzamento Chiuso (Chaining)

Anche conosciuto come Hashing Aperto, è un insieme di tecniche per le quali una chiave è sempre salvata nel secchio individuato dalla funzione di hashing.

Le collisioni sono gestite usando una struttura dati separata per ogni secchio.

Noi come struttura dati separata utilizzeremo delle liste concatenate, definite per il nostro progetto nel file python ***_LinkedList.py***.

2.1 Complessità Algoritmica

I limiti asintotici delle complessità temporale e spaziale previste per le operazione di ricerca, inserimento e cancellazione sono descritti nella seguente tabella:

Operazione	Caso Peggiore	Caso Medio	Caso Migliore
Ricerca	O(n)	O(1)	O(1)
Inserimento	O(n)	O(1)	O(1)
Cancellazione	O(n)	O(1)	O(1)
Spazio	O(m + n)		

Table 1: Complessità Temporale e Spaziale dell' hashing a indirizzamento chiuso

Dove m è la dimensione della tabella hash e n è il numero degli elementi inseriti.

3 Hashing con il Metodo delle Divisioni

3.1 Definizione della struttura dati

La struttura dati dell' hashing con il metodo delle divisioni è così definita:

```
class HashDivisionMethod:
    [uint]                  m
    [LinkedList()[]]        h
    [uint]                  size
```

3.2 Definizione della funzione di hashing

Il metodo delle divisioni mappa una chiave *key* k in uno degli m secchi, prendendo il resto di k diviso per m.

La funzione di hashing è così definità:

$$h(k) = key \mod m$$

3.3 Implementazione software

L'implementazione dell'hashing con il metodo delle moltiplicazioni è descritta nei file Python ***HashMoltiplicationMethod.py*** e ***_HashTable.py*** alla relazione allegati.

I seguenti metodi sono ereditati dalla classe **_HashTable** :
insert(key, value) search(key) remove(key) is_full() load_factor() .

La classe **HashDivisionMethod** implementa invece la funzione di hash:
h_ix(key)

4 Hashing con il Metodo delle Moltiplicazioni

4.1 Definizione della struttura dati

La struttura dati dell' hashing con il metodo delle divisioni è così definita:

```
class HashMultiplicationMethod:
    [uint]                  m
    [LinkedList()[]]        h
    [uint]                  size
    [uint]                  A
```

4.2 Definizione della funzione di hashing

Il metodo delle moltiplicazioni moltiplica una chiave *key* k per una costante A tale che $0 < A < 1$, ne prende la parte frazionaria, la moltiplica per m e infine ne estrae la parte intera.

La funzione di hashing è così definità:

$$h(k) = \lfloor m * ((key * A) \ mod 1) \rfloor$$

4.3 Implementazione software

L'implementazione dell'hashing con il metodo delle divisioni è descritta nei file Python ***HashDivisionMethod.py*** e ***_HashTable.py*** alla relazione allegati.

I seguenti metodi sono ereditati dalla classe **_HashTable** :
`insert(key, value)` `search(key)` `remove(key)` `is_full()` `load_factor()` .

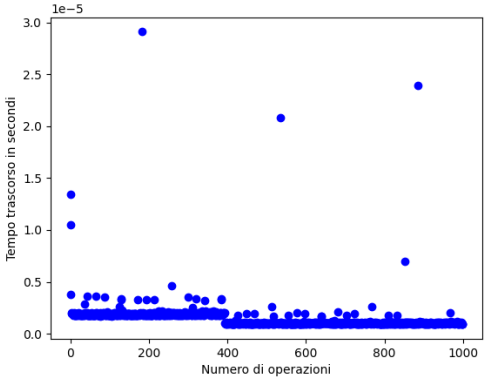
La classe **HashDivisionMethod** implementa invece la funzione di hash:
`h_ix(key)`

5 Test sulle operazioni

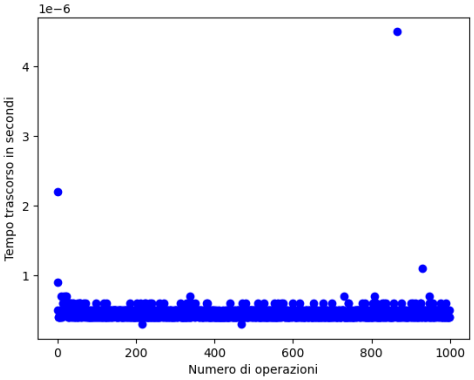
Di seguito vengono riportati i grafici eseguiti sull'implementazione software dell' hashing con il metodo delle divisioni e con il metodo delle moltiplicazioni, attraverso l' unità di testing definita nel file Python `TestUnit.py` ed eseguita da `main.py` per le operazioni di inserimento, ricerca, cancellazione e per misurare il numero delle collisioni.

I test sono stati eseguiti per una sequenza in input *Randomized*. La dimensione della tabella hash è definita costante a `m = 1000`

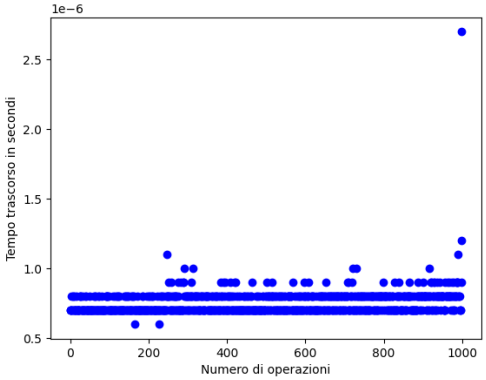
5.1 Operazioni con il Metodo delle Divisioni



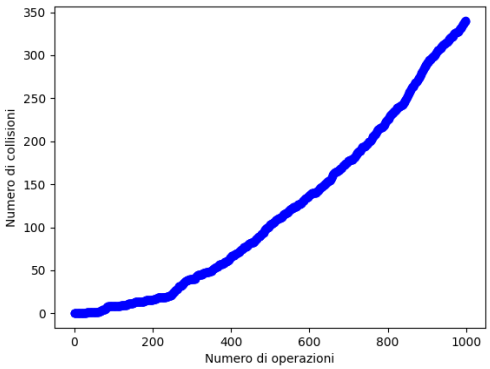
(a) Inserimenti



(b) Ricerche

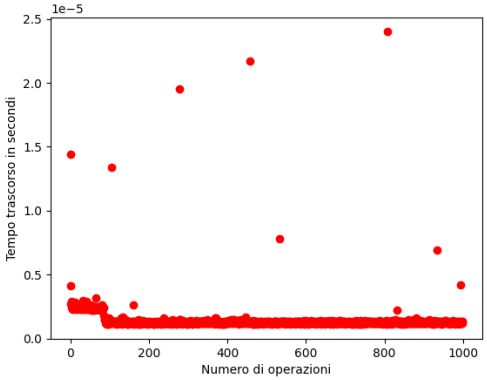


(c) Eliminazioni

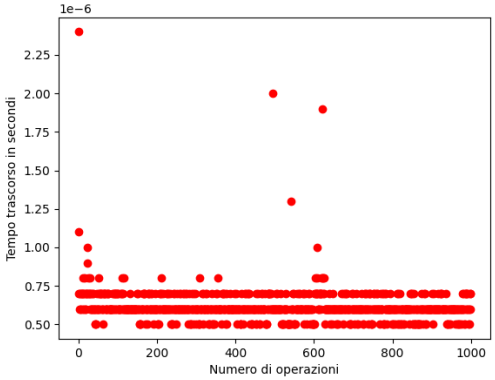


(d) Collisioni in inserimento

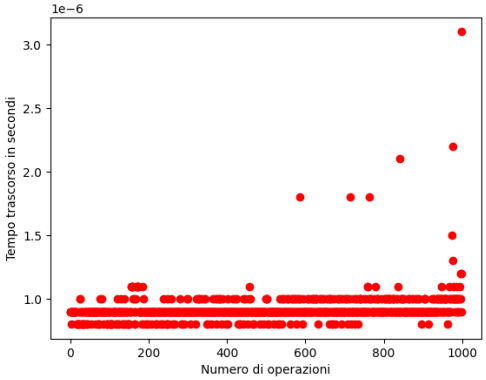
5.2 Operazioni con il Metodo delle Moltiplicazioni



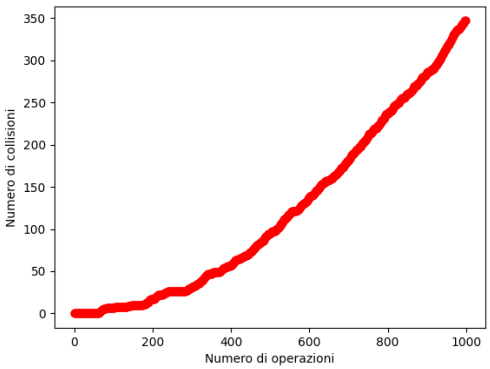
(a) Inserimenti



(b) Ricerche

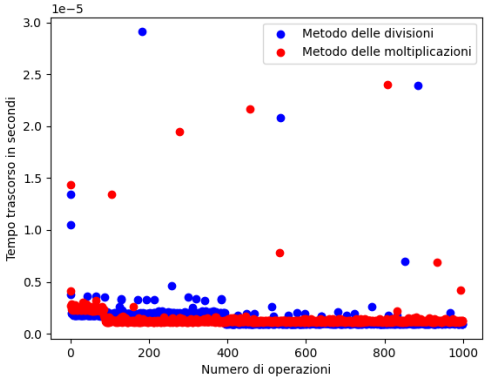


(c) Eliminazioni

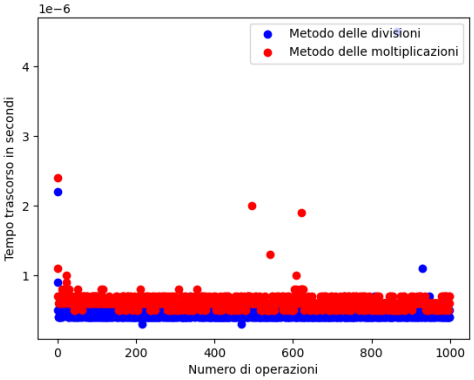


(d) Collisioni in inserimento

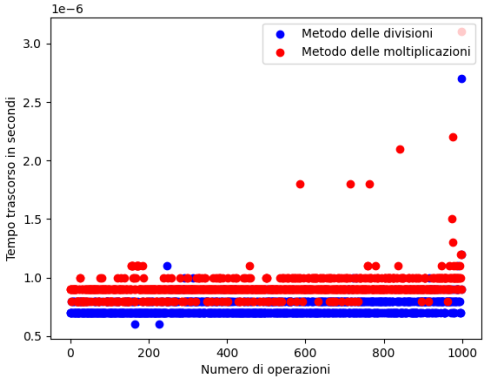
5.3 Confronto tra le operazioni per i due metodi



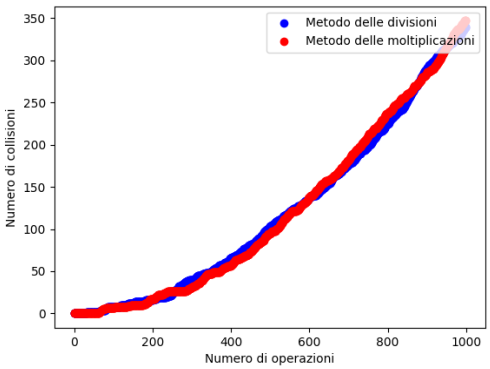
(a) HDM vs HMM Inserimenti



(b) HDM vs HMM Ricerche



(c) HDM vs HMM Eliminazioni



(d) HDM vs HMM Collisioni in Inserimento

6 Conclusioni

I risultati test effettuati coincidono con la complessità attesa delle operazioni di inserimento, ricerca e eliminazione per le tabelle hash che utilizzano il metodo delle divisioni e il metodo delle moltiplicazioni.

È osservabile un piccolo miglioramento di prestazione per le operazioni di inserimento, ricerca e eliminazione, utilizzando il metodo delle divisioni (HDM). È inoltre osservabile una riduzione del numero di collisioni per l'operazione di inserimento, utilizzando il metodo delle moltiplicazioni (HMM).