ALGORITMI I

Relazione sulle tabelle di hash

**Descrizione del progetto**

Il progetto si concentra sullo studio delle tabelle di hash,ovvero una variante della tabella ad accesso diretto,che calcola la posizione dell'elemento da inserire in tabella mediante una funzione h(k).

Le tabelle di hash sono state introdotte per sopperire allo spreco di memoria dovuto alla tabella ad accesso diretto, e a differenza di quest'ultime,nelle tabelle di hash è possibile anche che k(l'elemento da inserire) non sia intero,tuttavia l'importante è che la funzione h(k) restituisca un intero.

Ciò comporta un notevole miglioramento del *fattore di carico,* ovvero il rappporto fra il numero degli elementi da inserire in tabella (*n),* e la dimensione della tabella (*m).*

Maggiore è il fattore di carico minore è lo spreco di memoria,tuttavia al costo di mantenere un fattore di carico sufficientemente alto,bisogna convivere con il problema delle *collisioni*: situazioni in cui due *h* differenti generano lo stesso *h(k)*.

Le collisioni possono essere gestite con:

* liste di collisione
* applicando una legge di scansionamento (es. lineare o hashing doppio)

per lo svolgimento del progetto è stato scelto l'hashing doppio.

**Scelte progettuali**

Il progetto richiede di implementare e provare una tabella di hash per indicizzare un archivio di studenti, ed eventualmente capire quale sia il suo fattore di carico ottimale.

Le operazioni richieste dal progetto sono:

* la lettura da file e l'inserimento degli studenti (80) nella tabella di hash
* la ricerca di uno o più studenti data la sua matricola
* la visualizzazione dei dati dello studente eventualmente trovato e del numero di hit

E’ stato deciso di riunire le funzioni utili al programma in un file a parte, sfruttando la compilazione separata. In **utils.c** è possibile trovare le funzioni “generali” usate dal programma e le funzioni per le tabelle di hash.

Si è poi creato un file **utils.h** contente tutti i prototipi delle funzioni inserite nel file precedentemente menzionato, ed infine si è poi realizzato il programma vero e proprio (**tabHash.c**).

All'inizio del programma viene controllato l'input,il quale si aspetta un file di lettura (*studenti.txt* in teoria) e il nome del file di output generato dal codice con estensione *.csv*.

Viene dichiarato un array *dim,* contenente le dimensioni *m* da testare,tutte numeri primi poiché ciò è richiesto dalla funzione di hashing.

Successivamente per ogni dimensione contenuta nell'array si svolgono le seguenti operazioni:

1. creazione della tabella di hash di lunghezza *m* e inizializzazione delle sue celle a NULL
2. lettura ed inserimento degli *n* studenti nella tabella
3. ricerca di *n* studenti casuali
4. stampa del fattore di carico corrente e della media di hit ottenuta
5. deallocazione della tabella

Per la ricerca è stata utilizzata una funzione *crea\_matricola,* che genera un numero di matricola casuale fra la minima presente nel file e la massima per testare la ricerca.

Dato l'enorme range,difficilmente la ricerca risulta essere positiva,tuttavia per ogni studente vengono stampati il numero di bucket visitati come richiesto.

**Grafico e considerazioni**

