Algoritmi e Strutture Dati

Progetto di Laboratorio Esercizio 2
Anno Accademico: 2022/2023

Abdelhamid Badrane[matricola: 902154]
abdelhamid.badrane@edu.unito.it
Marika Torrente[matricola: 916315]
marika.torrente@edu.unito.it

Sommario

1.1 Introduzione

L'esercizio 2 richiede l'implementazione di una libreria che realizza la skiplist in cui vengono caricati i dati di un file dizionario, che successivamente vengono confrontati con le parole contenute all'interno di un file correctme.txt contenenti possibili parole errate con lo scopo di rintracciare e stampare a video le parole scritte non correttamente, ovvero quelle che non sono presenti all'interno del dizionario.
La skiplist è una struttura dati che memorizza una lista ordinata di elementi ed è gestita in base ad un'altezza max_height, dalla quale viene calcolata un'altezza randomica estratta per ciascun nodo tale che non superi max_height, permettendo di velocizzare le operazioni di ricerca su un insieme di natura lineare.

1.2 Guida all'utilizzo

Comandi make per l'utilizzo del programma:

-make all: compila e crea l'eseguibile per il main e gli unit test
-make main: compila il file main.c
-make compile_test: compila il file relativo agli unit test
-make test: compila, crea ed esegue il programma per gli unit tests

-make run: controlla la presenza dei parametri aggiuntivi richiesti, corrispondenti alla presenza dei file correctme.txt, dictionary.txt e max_height, se non sono presenti ne segnala la mancanza e nel caso contrario viene avviato il main

Si è scelto volutamente un doppio controllo per quanto riguarda i parametri richiesti, questo controllo viene effettuato sia nel makefile che all'avvio del main, in modo da permettere esecuzione anche tramite terminale.

1.3 Funzionalità del Main

La prima funzionalità di cui si occupa il main è quella di controllare la presenza dei tre parametri passati al main, in caso i parametri non siano presenti il programma viene terminato con esito failure. In caso siano presenti e corretti l'esecuzione continua e viene avviata la funzione find_errors(), che si occupa di popolare la skiplist con le parole presenti all'interno del file dizionary.txt; ovvero il file contenente le parole corrette.
Una volta effettuato il caricamento si provvede subito alla ricerca delle singole parole presenti nel file correctme.txt, eliminando opportunamente la punteggiatura e lettere maiuscole sostituendole con minuscole nel caso non fossero presenti, se le parole non sono presenti all'interno del dizionario vengono considerate errate e stampate a video.

1.4 Tempi di caricamento

Durante l'esecuzione di diversi test, che si differenziano dall'altezza della skiplist espressa dalla variabile max_height si può constatare un impatto sul caricamento dei dati, la velocità di esecuzione risulta diminuire con l'aumento dei puntatori, la tabella si limita ad H=20 data la poca differenza con H=15, si ipotizza che dopo il 20 i numeri di caricamento saranno sempre meno differenti

MAX_HEIGHT	Tempo di caricamento
H=5	$\sim 406\ sec$
H=7	$\sim 58,94\ sec$
H=10	$\sim 6,34\ sec$
H=15	$\sim 2,3\ sec$
H=20	$\sim 2,25\ sec$
H=30	$\sim 2,24\ sec$

1.5 Tempi di correzione

I tempi di ricerca sono tutti minori di 1 secondo, le differenze tra l'una e l'altra sono quasi impercettibili, ma pur sempre visibili, questioni di millesimi di secondi.

1.6 Conclusioni

Come è possibile constatare dai confronti effettuati l’altezza massima max_height della skiplist ha un effetto sui tempi di caricamento e di ricerca della skiplist, più è grande l’altezza meno è il tempo di esecuzione.