

## Esercizio 1

Scrivere un programma che implementi una tavola di hash (2,1) (cioè con 2 funzioni di hash e 1 elemento per secchio). Il programma permette di inserire e cercare elementi nella tavola usando il metodo spiegato a lezione. Le chiavi da inserire sono numeri interi positivi o al più zero. Le funzioni di hash da utilizzare sono della forma

$$h(x) = ((ax + b) \bmod p) \bmod n$$

dove  $p$  è un primo,  $n$  è il numero di secchi e  $a$  e  $b$  possono essere scelti casualmente (attenzione a mantenere  $a > 0$ ). Alcuni primi tra cui scegliere: 131071, 225749, 499481, 699757, 899051.

L'input è formato da una prima riga che contiene  $N$ , il numero di secchi da utilizzare. Il programma entra poi in un ciclo infinito che legge ogni volta 2 interi: il tipo di operazione e il suo argomento. L'operazione "0" è l'operazione di inserimento, ed il suo argomento è il numero da inserire. L'output associato ad una operazione di inserimento è: -1 se l'inserimento fallisce, 0 se l'elemento era già presente o il valore  $x$ , dove  $x$  indica il numero di scambi che sono eseguiti durante la procedura di inserimento.

L'operazione "1" è quella di ricerca, dove l'argomento indica l'elemento da cercare. L'output dell'operazione "1" è -1 se l'elemento non è presente o il secchio in cui si trova.

## Esercizio 2

Estendendo l'esercizio 1 per calcolare la frequenza dei termini in un documento. In un documento, ogni stringa rappresentante una parola è un termine. Alcuni termini sono ripetuti, come ad esempio la parola "operazione" nel testo dell'esercizio 1, che compare 6 volte. Il vostro programma deve leggere un testo e produrre un elenco di tutti i termini che vi compaiono, con associata la loro frequenza. Ogni termine deve apparire in questo elenco una ed una sola volta e tutte le frequenze devono partire da 1. Potete assumere che vi verranno date soltanto parole minuscole separate da uno spazio (no punteggiatura o altro).

L'input è formato da una prima riga contenente  $N$  e  $M$ , rispettivamente il numero di secchi da usare nella tavola di hash e  $M$ , il numero di parole del testo. Seguono  $M$  stringhe separate da spazio, tutte di lunghezza inferiore a 1024.

L'output è formato da una prima riga contenente  $U$ , il numero di termini univoci contenuti nel testo, seguito da  $U$  righe. Ogni riga contiene un termine differente seguito dalla sua frequenza nel testo originario. Notate che la somma delle frequenze deve sempre fare  $M$ . Nel caso la tavola di hash sia troppo piccola e l'inserimento fallisca, stampare semplicemente  $0 \setminus n$ .

Utilizzate il file “hashstringhe.c” per ottenere le 2 funzioni di hash per stringhe. Inoltre, sempre nel file zip trovate tre testi (testo0.txt testo1.txt e testo2.txt) per provare il vostro programma. Sperimentate vari valori di  $N$  per riuscire a impedire l’inserimento nella tavola di hash e/o misurate il tempo di inserimento e cancellazione con  $N$  alto.

### **Esercizio 3**

Estendete l’esercizio 1 (ed eventualmente l’esercizio 2) per utilizzare un  $(2, P)$ -schema con  $P > 1$ , cioè dove ogni secchio può contenere fino a  $P$  elementi. Pertanto, potranno convivere fino a  $P$  elementi con gli stessi hash all’interno di un certo secchio. In questo modo, gli input file degli esercizi 1 e 2 avranno un parametro in più sulla prima riga:  $P$ . Confrontate le velocità e le capacità di carico di questa nuova tabella, anche solo con  $P = 3$ , con la vostra vecchia implementazione.