

Laboratorio

3 Maggio 2012

Tabella Hash: Inserimento

Scrivere un programma che legga da tastiera una sequenza di n interi **distinti** e li inserisca in una tabella hash di dimensione $2n$ posizioni utilizzando liste monodirezionali per risolvere eventuali conflitti. Utilizzare la funzione hash $h(x) = ((ax + b) \% p) \% 2n$ dove p è il numero primo 999149 e a e b sono interi positivi minori di 10.000 scelti casualmente. Una volta inseriti tutti gli interi, il programma deve stampare il numero totale di conflitti e la lunghezza massima delle liste.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

Prima di scrivere il programma chiedersi perché la tabella ha dimensione $2n$ e non n .

Tabella Hash: Inserimento con eliminazione dei non distinti

Scrivere un programma che legga da tastiera una sequenza di n interi **NON distinti** e li inserisca senza duplicati in una tabella hash di dimensione $2n$ posizioni utilizzando liste monodirezionali per risolvere eventuali conflitti. Utilizzare la funzione hash $h(x) = ((ax + b) \% p) \% 2n$ dove p è il numero primo 999149 e a e b sono interi positivi minori di 10.000 scelti casualmente. Una volta inseriti tutti gli interi, il programma deve stampare il numero totale di conflitti, la lunghezza massima delle liste e il numero di elementi distinti.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

Liste: Ordinamento

Scrivere un programma che legga da tastiera una sequenza di n interi, li inserisca in una lista bidirezionale e li ordini utilizzando **Insertion sort**. L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

Liste: Cancellazione

Scrivere un programma che legga da tastiera una sequenza di n interi distinti e li inserisca in una lista monodirezionale. Successivamente il programma deve calcolare la media aritmetica dei valori della lista e eliminare tutti gli elementi il cui valore è inferiore alla media.

Avere cura di liberare la memoria dopo ogni cancellazione.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

Liste: Somme Suffisse

Scrivere un programma che legga da tastiera una sequenza di n interi maggiori di 0 e li inserisca in una lista nell'ordine di immissione. La lista deve essere **monodirezionale**. Successivamente il programma deve sostituire il valore di ciascun elemento con la somma dei valori degli elementi che lo seguono nella lista. (Suggerimento: si utilizzi la ricorsione per ottenere la somma ad ogni passo.)

Osservare che tutti gli interi nella nuova lista sono ora sicuramente distinti. Il programma entra in un ciclo infinito nel quale legge un intero i da tastiera e lo cerca nella lista risultante. Se i si trova nella lista, stampa la sua posizione (contando da 0), altrimenti stampa *NO*.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

La lista iniziale contiene i valori 1 2 4 2 2 1 4 1

La lista dopo il primo passo contiene i valori 16 14 10 8 6 5 1 0

Prova di esame 15 Giugno 2011

Sono date in input due sequenze di stringhe. La prima consiste di N stringhe che devono essere inserite in una tabella hash. La tabella dovrà avere dimensione $2N$ e la funzione hash da utilizzare è definita come segue:

$$h(s) = ((4321v + 5678) \% 999149) \% 2N$$

dove v è un **intero** che rappresenta la somma di tutti i caratteri che compongono la stringa s . Per poter effettuare la somma si deve utilizzare il cast ad intero di ogni singolo carattere:

```
int v = 0, i;  
for(i=0; s[i]; i++) v += (int)s[i];
```

La seconda sequenza consiste di M stringhe che vanno ricercate all'interno della tabella precedentemente creata. Per ogni stringa s di questa sequenza, si richiede in output il numero delle stringhe che sono alfabeticamente più grandi di s fra quelle contenute nello slot di s della tabella hash.

L'input è formattato nel seguente modo. Nella prima riga è contenuto l'intero N . Seguono poi N righe contenenti ognuna una stringa appartenente alla prima sequenza. Dopo l'ultima stringa della prima sequenza, viene passata in input una riga contenente l'intero M , seguita da M righe contenenti le stringhe della seconda sequenza. Le stringhe delle due sequenze sono costituite solo da caratteri alfanumerici (a-Z e 0-9, senza spazi) e lunghe al massimo 100 caratteri e minimo un carattere.

L'output invece è costituito da M righe ognuna delle quali contiene un intero che rappresenta il risultato della ricerca come specificato di sopra.

Potete ovviamente avvalervi di funzioni di libreria standard, come `strcmp`, `strlen`, ecc.

Esempio

Input	baab
	paperino
4	
pippo	
caaa	
topolino	
abba	
2	

Output

1
0

Puzzled

Griglia infetta

Abbiamo una griglia 8×8 e la possibilità di infettare 7 delle sue celle. Ad ogni intervallo di tempo, vengono infettate tutte le celle che hanno almeno due lati infetti. Il processo termina non appena non sono più possibili nuove infezioni. Mostrare come scegliere le prime 7 celle in modo che alla fine del processo l'intera griglia sia infetta, o dimostrare che una tale configurazione non può esistere.