Fondamenti di Informatica

Prof. M. Gavanelli, E. Lamma

14 Giugno 2018

Esercizio (Punti 20) (1h e 45 min)

Due file di tipo di testo, `mara.txt`, `lucia.txt`, contengono l'elenco di reperti archeologici catalogati da due studentesse, Mara e Lucia. Per ogni reperto, nel file, su una linea è riportata una parola al massimo di 30 char che lo descrive (ad esempio, anfora, brocca, etc), lo stato del reperto (intero con valore 1 se in buono stato, 0 se gravemente danneggiato), e profondità del ritrovamento (in metri, un intero). I file non sono ordinati e ogni parola può ripetersi nel file.

Si realizzi un programma C, organizzato in **almeno** tre funzioni, rispettivamente dedicate a:

- costruire in memoria centrale una lista L, che riporta tutti i reperti dei due file, ordinata sul campo parola.

  La **funzioneA** riceve come parametri:

  - il puntatore a un file e
  - il puntatore alla radice della lista data in ingresso

  (più eventuali parametri a scelta) e restituisce il puntatore radice della lista L modificata; la **funzioneA** va quindi invocata due volte dal main, una volta passando il file `mara.txt`, e un'altra passando il file `lucia.txt`;

- mostrare a video il contenuto (ordinato) di L; la **funzioneB** riceve come parametri:

  - il puntatore a L,

  più eventuali parametri a scelta, e restituisce `void` ;

- accedendo a L, contare quanti sono i reperti con parola "anfora" e stampare il valore calcolate su un file di uscita, **parola.txt**. Il file **parola.txt** va consegnato con il codice sorgente. Questa funzione (**funzioneC**) riceve:

  - il puntatore radice L,
  - il puntatore al file di uscita,

  più eventuali parametri a scelta, e restituisce `void`.

- accedendo a L, contare quanti sono i reperti trovati a profindità maggiore di 3 metri, e stampare i valori calcolati su un file di uscita, **profondi.txt**. Il file **profondi.txt** va consegnato con il codice sorgente. Questa funzione (**funzioneC**) riceve:
  - il puntatore radice L,
  - il puntatore al file di uscita,

  più eventuali parametri a scelta, e restituisce `void`.

Per chi svolge il compito A+B (Ulteriori punti 20 su tot. di 60 per A+B; tempo +45 min; tot 2,5 h):

Si scriva una funzione (**funzioneAB**) per riempire un array di 100 elementi, V, in cui ciascun elemento è costituito da una parola (stringa al massimo di 30 char). Sapendo che il file `mara.txt` ha più di 100 reperti, la funzione inserisce nel vettore V le prime 100 parole del file.

Questa funzione (**funzioneAB**) riceve come parametri il vettore V e il puntatore al file `mara.txt` più eventuali parametri a scelta, e restituisce `void`.

Nel `main`, si ordini il vettore V tramite una opportuna chiamata della funzione `qsort` e si stampi poi su un file di testo `outputAB.txt` il contenuto dell'array V.

NOTA BENE: Si consegnino i sorgenti e il file di uscita generato (per A+B, consegnare i due file di uscita generati). È possibile utilizzare **librerie C** (ad esempio per le stringhe). Nel caso si strutturi a moduli l'applicazione qualunque **libreria utente** va riportata nello svolgimento.

Esercizio 2 (Punti 3 su 31) (15 min)

Sia data la seguente funzione `fun` che riceve un carattere e un albero binaro di caratteri

```
int fun(char i, tree T)
{ if (T==NULL) return 0;
  else  if (i==T->value) return 1+fun(i,T->left)+fun(i,T->right);
        else
            return 1+fun(i,T->left)+fun(i,T->right);
}
```

Si indichi cosa fa la funzione `fun` e se ne valuti la complessità asintotica come numero di test `i==T->value`. Come varia la complessità varia se l'albero è binario di ricerca?

NOTA BENE: Si consegni la soluzione in un file `teoria.txt`.