Algoritmi e Strutture di Dati – A.A. 2011-2012
Seconda Prova dell'appello del 3 e 4 settembre 2012
Libri e appunti chiusi

☐ Studente 270/04 (tempo prova = 2:15 h) ☐ Studente 509/99 (tempo prova = 2:00 h)

☐ Ho bisogno di una correzione veloce in quanto _____

_____

Cognome: _____ Nome: _____ Matricola:_____

PSEUDOCODIFICA

Negli esercizi seguenti supponi che una lista L semplicemente concatenata contenga un campo `first` che è un riferimento al primo elemento, dove ogni elemento ha il campo `next` e il campo `info` (un intero). Un grafo diretto, invece, è rappresentato con un array A in cui ogni elemento A[*u*] è un riferimento al primo elemento della lista di adiacenza doppiamente concatenata del nodo *u* (con i campi `prev`, `info` e `next`).

Esercizio 1

Scrivi lo pseudocodice della procedura INSERISCI(L,*n*) che prende in input una lista L di interi (semplicemente concatenata) ed un intero *n* ed inserisce un elemento in testa alla lista L con il valore del campo `info` pari ad *n*.

Esercizio 2

Scrivi lo pseudocodice della procedura TRASFORMA(A) che prende in input un array di interi A (di lunghezza A.length) e produce in output una lista L semplicemente concatenata di interi ordinata in modo non decrescente che contenga tutti gli elementi di A.

Esercizio 3

Scrivi lo pseudocodice della procedura ROVESCIA(L) che prende in input una lista L semplicemente concatenata e ordinata in modo non decrescente e produce in output una lista semplicemente concatenata ordinata in modo non crescente con gli stessi valori (L non deve essere necessariamente preservata).

Esercizio 4 (solo studenti D.M. 270/04)

Scrivi lo pseudocodice della procedura CONTA_RAGGIUNTI(G,*n*,L) che prende in input un grafo G, l'indice *n* di un nodo e una lista semplicemente concatenata L di indici di nodi e conta quanti nodi i cui indici sono contenuti in L sono raggiungibili con un cammino che parte dal nodo con indice *n*.

Esercizio 5

Discuti la complessità computazionale (nel solo caso peggiore) delle procedure che hai proposto per gli esercizi precedenti, utilizzando *n* per denotare il numero totale dei nodi dell'albero o della lista.

CODIFICA C

Si immagini di dover gestire un indice analitico di un libro. Ogni elemento di questo indice è una coppia (*p*,*n*) dove *p* è una parola (stringa di caratteri lunga max 20) mentre *n* è il numero di pagina del libro in cui trovare la parola. Tale indice è ordinato in modo alfabetico rispetto a *p*. La lunghezza dell'indice analitico non è nota (non è possibile in alcun modo fissare la sua lunghezza massima). Si assuma che nell'indice ogni parola *p* possa ricorrere solo una volta. Esempio:

```
albero, 103
automa, 34
biconnesso, 57
grafo, 134
```

Implementare in linguaggio C quanto segue:

Esercizio 1 (10%)

Le strutture dati più adeguate anche in riferimento ai metodi dei punti seguenti da implementare

Esercizio 2 (35%)

Un metodo `INSERISCI` che dato l'indice *i*, la parola *p* e il numero *n*, inserisca in *i* il nuovo elemento (*p*,*n*) mantenendo l'ordine alfabetico (si assuma che *p* non esista in *i*).

Con riferimento all'esempio, dato l'indice *i*, e la coppia `(codice,43)` otteniamo il risultato

```
albero, 103
automa, 34
biconnesso, 57
codice, 43
grafo, 134
```

Esercizio 3 (30%)

Un metodo `CERCA` che dato l'indice *i*, la parola *p* e il numero *n* restituisca 1 se la parola *p* esiste nell'indice, 0 altrimenti. Inoltre, nel caso in cui *p* non esista in *i*, inserisce (*p*,*n*) nell'indice. Considerando lo stesso esempio qui sopra, dato l'indice *i*, e la coppia `(grafo, 134)` in questo caso otteniamo 1. Se avessimo `(codice, 43)` otterremo 0 e la modifica dell'indice come nel punto precedente.

Esercizio 4 (25%)

Un metodo ricorsivo `CREAFILE` che dato un carattere *c*, l'indice *i*, crei un file di testo "`parole.txt`" che contenga tutte le parole dell'indice *i* che iniziano con il carattere *c* (una parola per ogni riga). Non è ammessa alcuna implementazione iterativa di `CREAFILE` (questa non sarà valutata in nessun modo). Solo metodi di supporto possono essere iterativi. Nel caso dell'indice dell'esempio e il carattere 'a', otteniamo il seguente contenuto del file:

```
albero
automa
```