
Algoritmi e Strutture di Dati – A.A. 2011-2012
Seconda Prova dell'appello del 12 e 13 giugno 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 un grafo diretto sia 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 . Un albero binario T ha invece il solo riferimento alla radice $T.root$ (ogni nodo dell'albero ha i campi `parent`, `left`, `right` e `info`).

Esercizio 1

Scrivi lo pseudocodice della procedura `FIGLI-RADICE(T)` che prende in input un albero T e restituisce **true** se la radice dell'albero ha due figli, **false** altrimenti (un solo figlio o nessun figlio).

Esercizio 2

Scrivi lo pseudocodice della procedura `CERCA(T,u)` che prende in input l'albero T e un valore intero u , e restituisce un riferimento al nodo che ha come valore `info` il valore u .

Esercizio 3

Scrivi lo pseudocodice della procedura `CAMMINO(A,u,v)` che prende in input un grafo A e verifica se esiste un cammino diretto dal nodo con indice u al nodo con indice v .

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

Scrivi lo pseudocodice della procedura `ANTENATO(T,u,v)` che prende in input un albero T e gli interi u e v e ritorna **true** se il nodo con etichetta u è un antenato del nodo con etichetta v , **false** altrimenti.

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 del grafo.

CODIFICA C

Un sillabario è un elenco di sillabe distinte, dove ogni sillaba è costituita da massimo 3 lettere (caratteri). Ad esempio il sillabario s contiene le sillabe $\{go,co,ste,ro,al,me,sa\}$. Una parola p è anch'essa una sequenza di sillabe, non necessariamente distinte. Per esempio $p = \{co,co,me,ro\}$. Diciamo che un sillabario s supporta una parola p se s contiene tutte le sillabe di p . Nell'esempio qui sopra il sillabario s supporta la parola p .

Si definisca in linguaggio C, con l'utilizzo delle sole librerie `<stdio.h>` e `<stdlib.h>` (quindi non è ammesso l'uso della libreria `<string.h>`), quanto segue:

Esercizio 1 (10%)

Le strutture dati per gestire una parola e un sillabario e un elenco di parole. Non sono date in alcun modo come input le lunghezze di parole, sillabari, ed elenchi di parole. Si ottimizzi l'implementazione di tali strutture rispetto a quanto richiesto negli esercizi che seguono.

Esercizio 2 (40%)

Una funzione che, dati una parola p e un sillabario s , restituisca il numero di sillabe di p che non sono contenute in s .

Esercizio 3 (25%)

Una funzione che, dato un elenco di parole e un sillabario, restituisca il numero di parole supportate dal sillabario.

Esercizio 4 (25%)

Una funzione ricorsiva che data una parola p stampi su di un file di testo di nome "parola.txt" tutte le sillabe di p , una per ogni riga. Ad esempio, data la parola $p = \{co,co,me,ro\}$ occorre stampare sul file:

```
co
co
me
ro
```

L'implementazione di una funzione non ricorsiva otterrà 1/4 della valutazione prevista per questo esercizio.

E' possibile implementare tutte le funzioni di supporto per la risoluzione dei sopracitati punti. E' vietato l'utilizzo di qualsiasi costrutto, funzione o procedura che non appartenga al linguaggio C.