

02MNO Algoritmi e Programmazione 01JKE APA I / 01JKF APA II

Appello del 06/02/2014 - Prova di teoria (12 punti)

1. (2 punti)

Si risolva la seguente equazione alle ricorrenze mediante il metodo dello sviluppo (unfolding):

$$T(n) = 8T(n/2) + n^3 \quad n \geq 2$$

$$T(1) = 1$$

2. (2 punti)

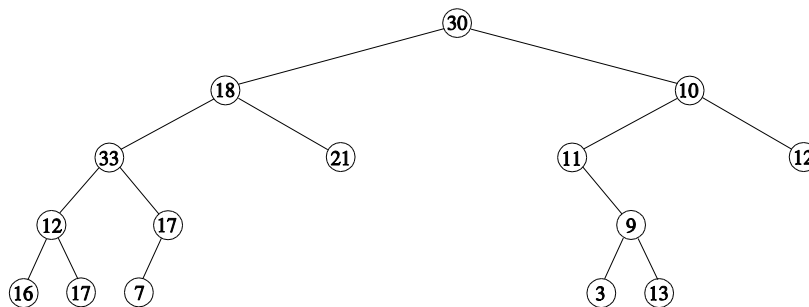
Sia data la sequenza di interi, supposta memorizzata in un vettore:

12 29 9 11 21 33 22 13 15 19 25 30

si eseguano i primi 2 passi dell'algoritmo di quicksort per ottenere un ordinamento ascendente, indicando ogni volta il pivot scelto. NB: i passi sono da intendersi, impropriamente, come in ampiezza sull'albero della ricorsione, non in profondità. Si chiede, pertanto, che siano ritornate le 2 partizioni del vettore originale e le due partizioni delle partizioni trovate al punto precedente.

3. (3 x 0.5 punti)

Si visiti in pre-order, in-order e post-order il seguente albero binario. In caso di chiavi ripetute, distinguere le istanziazioni con pedici.

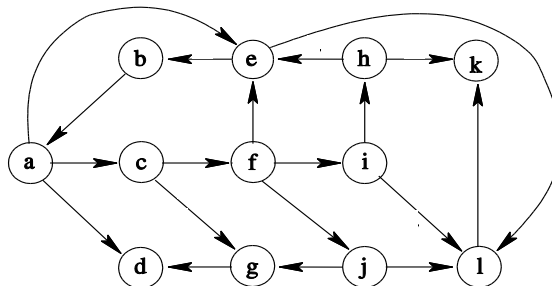


4. (2 punti)

Sia data la sequenza di chiavi $S_1IDE_1E_2F_1F_2E_3CTS_2$, dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto ($A=1, \dots, Z=26$) con eventuale pedice. Si riporti la struttura di una tabella di hash di dimensione 23, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza indicata. Si supponga di utilizzare l'open addressing con double hashing utilizzando le funzioni di hash $h_1(k) = k \bmod 23$, $h_2(k) = 1 + (k \bmod 21)$.

5. (2.5 punti)

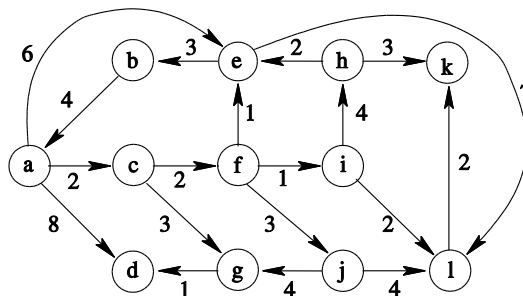
Dato il seguente grafo orientato:



se ne determinino mediante l'algoritmo di Kosaraju le componenti fortemente connesse. Si consideri **a** come vertice di partenza e, qualora necessario, si trattino i vertici secondo l'ordine alfabetico.

6. (2 punti)

Sul seguente grafo orientato e pesato, si determinino i valori di tutti i cammini minimi che collegano il vertice **a** con ogni altro vertice mediante l'algoritmo di Dijkstra. Si assuma, qualora necessario, un ordine alfabetico per i vertici e gli archi.



02MNO Algoritmi e Programmazione 01JKE APA I / 01JKF APA II

Appello del 06/02/2014 - Prova di programmazione (18 punti)

L'**aritmetica verbale** è un gioco matematico pubblicato per la prima volta da Henry Dudeney nel numero del luglio 1924 della rivista Strand Magazine. Si supponga di avere dei numeri interi le cui cifre sono "criptate" sotto forma di lettere e di conoscere, sempre in forma criptata, il risultato di un'operazione aritmetica tra 2 interi criptati, come nel seguente esempio:

$$\begin{array}{rcccccc} & S & E & N & D & + \\ & M & O & R & E & = \\ \hline M & O & N & E & Y & \end{array}$$

Il gioco consiste nell'identificare per ciascuna lettera la cifra decimale tale per cui sia soddisfatta l'operazione. Nell'esempio considerato con la seguente corrispondenza

$$O = 0, M = 1, Y = 2, E = 5, N = 6, D = 7, R = 8 \text{ e } S = 9$$

si ottiene infatti

$$\begin{array}{rcccccc} & 9 & 5 & 6 & 7 & + \\ & 1 & 0 & 8 & 5 & = \\ \hline 1 & 0 & 6 & 5 & 2 & \end{array}$$

Ogni lettera presente è associata a una e una sola cifra decimale. Per semplicità si ipotizzi che l'unica operazione ammessa sia quella di somma tra 2 operandi interi criptati.

Assunzioni:

- le stringhe contengono solo caratteri alfabetici tutti maiuscoli o tutti minuscoli
- la cifra più significativa non può corrispondere allo 0
- le prime due stringhe non hanno necessariamente la stessa lunghezza, la terza stringa ha lunghezza coerente con quella delle prime 2, data l'operazione di somma
- le prime 2 stringhe hanno una lunghezza massima pari a 8, la terza coerente con quelle delle prime due
- nelle stringhe non compaiono più di 10 lettere distinte.

Si scriva un programma C che, dopo aver letto da tastiera le tre stringhe di caratteri, funzioni alternativamente come:

- verificatore di una soluzione fornita dall'utente da tastiera
- generatore automatico di una soluzione.

Nel primo caso, acquisita da tastiera in un formato a scelta la soluzione dell'utente, ne verifichi la correttezza. Nel secondo caso il programma calcoli e visualizzi una corrispondenza lettere-cifre decimali corretta.

Suggerimenti:

- il problema di identificare una corrispondenza lettere-cifre che soddisfa l'operazione si può risolvere con ricorsione con backtrack
- la condizione di accettazione della soluzione può essere verificata (senza pruning preliminare) una volta raggiunta la foglia dell'albero delle ricorsioni
- è possibile, in fase di accettazione, realizzare l'operazione di somma lavorando su singole cifre e propagando riporti oppure mediante conversione da vettore di cifre simboliche a intero e somma tra interi.

02MNO Algoritmi e Programmazione 01JKE APA I / 01JKF APA II

Appello del 06/02/2014 - Prova di programmazione (12 punti)

1. (2 punti)

Sia dati 3 vettori ordinati di interi. Sia nota la lunghezza di ciascuno di essi. Si realizzi una funzione C `merge3` che ritorni come risultato il vettore ordinato di interi che deriva dalla fusione dei 3 vettori di ingresso. Il prototipo della funzione è:

```
int *merge3 (int *a, int *b, int *c, int na, int nb, int nc);
```

dove `a`, `b` e `c` sono i 3 vettori già ordinati, `na`, `nb` e `nc` le loro lunghezze. Il vettore ritornato, di dimensione `na+nb+nc`, va allocato dinamicamente e generato mediante un'unica operazione di merge simultanea sui tre vettori. In altri termini è vietato fondere i primi 2 e poi il risultato con il terzo.

2. (4 punti)

Si scriva una funzione C in grado di inserire in una lista ordinata alcuni dati anagrafici di una persona, dati da una coppia cognome e nome, entrambe stringhe di caratteri di lunghezza massima 20. La chiave di ordinamento sia il cognome e, in caso di cognomi identici, il nome. La funzione, di cui il prototipo è:

```
int inserisciInOrdine (lista_t *lista, char *cognome, char *nome);
```

riceve una lista (tipo `lista_t`, corrispondente a un ADT di prima categoria), le due stringhe corrispondenti a cognome e nome, ritorna, come intero, un valore vero nel caso di inserimento corretto, falso nel caso di dato già presente. Si definiscano il tipo `lista_t` e il tipo per il generico dato in lista, tenendo conto che, per ogni persona, vanno generati per la lista duplicati allocati dinamicamente (e separati) per cognome e nome. Il codice deve essere scritto esplicitamente, in altri termini è vietato ricorrere a funzioni di libreria.

3. (6 punti)

Sia dato un vettore di float di lunghezza nota `n`. Ogni float rappresenta un movimento su un conto bancario: se positivo, è un'entrata, se negativo un'uscita. Si assuma che tutti i movimenti siano distinti. Dato un ordine per i movimenti, per ogni movimento, si definisce saldo corrente il valore ottenuto sommando algebricamente al saldo precedente (inizialmente 0) l'importo dell'operazione. Per ogni ordine di movimenti esisterà un saldo corrente massimo e un saldo corrente minimo, mentre il saldo finale sarà ovviamente lo stesso, qualunque sia l'ordine.

Si scriva una funzione C che, utilizzando un algoritmo ricorsivo, determini l'ordinamento del vettore tale da minimizzare la differenza tra saldo corrente massimo e saldo corrente minimo. La funzione deve ritornare la sequenza ordinata sul vettore di partenza.

PER ENTRAMBE LE PROVE DI PROGRAMMAZIONE (18 o 12 punti):

- *indicare nell'elaborato e nella relazione (oltre a nome, cognome e numero di matricola) anche il nome del corso per cui si sta sostenendo l'esame (AP, APA I+II).*
- *È consentito utilizzare chiamate a funzioni standard, quali ordinamento per vettori, inserzione/estrazione/ricerca relative a FIFO, LIFO, liste, BST, tabelle di hash e altre strutture dati, considerate come librerie esterne. Gli header file delle librerie utilizzate devono essere allegati all'elaborato. Le funzioni richiamate, inoltre, dovranno essere incluse nella versione del programma allegata alla relazione.*
- *Consegna delle relazioni (per entrambe le tipologie di prova di programmazione): entro martedì 11/02/2014, alle ore 24:00, via e-mail all'indirizzo: danilo.vendramineto@polito.it, usando come subject (oggetto) la stringa APA#<m>, essendo <m> il proprio numero di matricola. L'allegato alla mail deve essere costituito da un unico file: un archivio compresso, contenente sia il codice corretto, sia la relazione (NO eseguibili). **QUALORA IL CODICE SPEDITO CON LA RELAZIONE NON COMPILI CORRETTAMENTE, VERRÀ APPLICATA UNA PENALIZZAZIONE.** Si ricorda che la valutazione del compito viene fatta senza discussione o esame orale, sulla base dell'elaborato svolto in aula. Non verranno corretti i compiti di cui non sarà stata inviata la relazione nei tempi stabiliti.*