

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

Appello del 24/02/2014 - Prova di teoria (12 punti)

### 1. (1 punto)

Sia data la seguente sequenza di coppie, dove la relazione i-j indica che il nodo i è adiacente al nodo j:  
3-10, 5-3, 1-5, 7-2, 3-8, 5-4, 0-9, 1-5, 8-9, 10-4

si applichi un algoritmo di on-line connectivity con quickunion, riportando a ogni passo il contenuto del vettore e la foresta di alberi al passo finale. I nodi sono denominati con interi tra 0 e 10.

### 2. (1 punto)

Si ordini in maniera ascendente mediante counting-sort il seguente vettore di interi:

3 4 5 2 6 5 4 2 8 1 4 1 8 0 4

Si indichino le strutture dati usate nei passi intermedi.

### 3. (2 punti)

Sia data una coda a priorità inizialmente vuota implementata mediante uno heap. Sia data la sequenza di interi e carattere \*: 10 22 29 71 \* \* 58 \* 34 9 31 \* \* 14 \* 41 27 18 dove ad ogni intero corrisponde un inserimento nella coda a priorità e al carattere \* un'estrazione con cancellazione del massimo. Si riporti la configurazione della coda a priorità dopo ogni operazione e la sequenza dei valori restituiti dalle estrazioni con cancellazione del massimo.

### 4. (2 punti)

Sia dato un albero binario con 12 nodi. Nella visita si ottengono le 3 seguenti sequenze di chiavi:

preorder	33	41	23	1	7	11	17	19	13	3	27	5
Inorder	23	41	7	1	11	33	13	19	17	27	3	5
postorder	23	7	11	1	41	13	19	27	5	3	17	33

Si disegni l'albero binario di partenza.

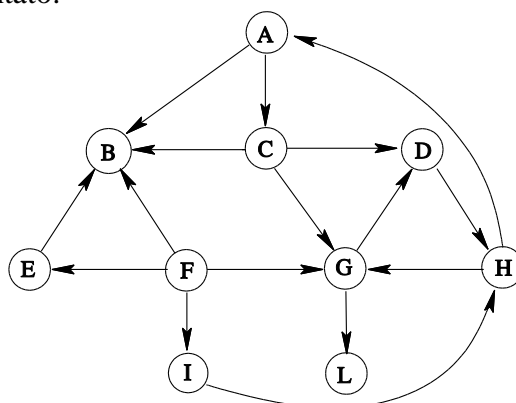
### 5. (1 punto)

Si supponga di aver memorizzato tutti i numeri compresi tra 1 e 1000 in un albero di ricerca binario e che si stia cercando il numero 531. Quali tra queste **non** possono essere le sequenze esaminate durante la ricerca? Perché?

500	550	510	540	533	539	532	531					
500	600	580	570	510	520	530	540	535	531			
570	100	200	300	400	500	660	550	520	525	530	531	
610	30	600	60	588	117	519	299	301	477	531		

### 6.

Sia dato il seguente grafo orientato:



- se ne effettui una visita in profondità, considerando A come vertice di partenza. Si etichettino indicando per ognuno di essi i tempi di scoperta e di fine elaborazione nel formato tempo1/tempo2 (2 punti)
- se ne effettui una visita in ampiezza, considerando a come vertice di partenza (1.5 punti)
- lo si ridisegni, etichettando ogni suo arco come T (tree), B (back), F (forward), C (cross), considerando a come vertice di partenza (1.5 punti).

Qualora necessario, si trattino i vertici secondo l'ordine alfabetico.

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

Appello del 24/02/2014 - Prova di programmazione (18 punti)

Un grafo orientato pesato è memorizzato su file mediante l'elenco dei suoi archi, con il seguente formato:

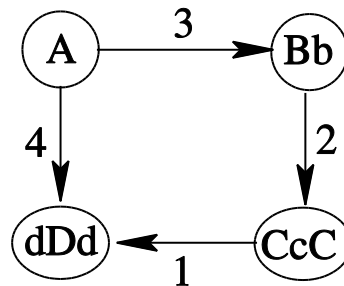
idV1 val idV2

che indica che il vertice con identificatore idV1 è connesso con un arco orientato di peso val (valore intero e positivo) al vertice idV2.

Si osservi che il numero di vertici del grafo è indefinito e che ciascun vertice è individuato mediante un identificatore alfanumerico di lunghezza massima uguale a 20 caratteri. Si può assumere che non esistano archi duplicati.

La figura seguente riporta un esempio di file e della relativa rappresentazione grafica del grafo memorizzato:

```
A 3 Bb
Bb 2 CcC
A 4 dDd
CcC 1 dDd
```



Si vuole scrivere un programma C in grado di:

- ricevere il nome di un file contenente la descrizione del grafo sulla riga di comando
- leggere il grafo e memorizzarlo in un'opportuna struttura dati
- verificare se il grafo è regolare o meno. Un grafo orientato si dice regolare se e solo se tutti i vertici hanno lo stesso `in_degree` e `out_degree` e questi sono uguali tra di loro
- effettuare un'iterazione all'interno della quale viene letto l'identificatore di un vertice:
  - se l'identificatore letto è uguale a "fine" il programma deve terminare
  - in caso contrario il programma deve determinare il cammino semplice con sorgente nel vertice indicato per il quale la somma dei pesi degli archi è massima. Per tale cammino si devono visualizzare gli archi attraversati (come coppia di identificatori alfanumerici dei vertici su cui insistono e peso dell'arco) e il peso totale.

Suggerimento:

Poiché il numero di vertici non è noto a priori, si suggerisce di calcolarlo tramite lettura preliminare del file e/o caricamento in una tabella di simboli.

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

Appello del 24/02/2014 - Prova di programmazione (12 punti)

### 1. (2 punti)

Sia dato un vettore di  $N$  interi. Si consideri il vettore rappresentato orizzontalmente con indice 0 a sinistra e indice  $N-1$  a destra. Una rotazione elementare del vettore verso destra (sinistra) consiste nello spostare tutti i dati di una posizione verso destra (sinistra), portando il dato di indice  $N-1$  (0) in posizione 0 ( $N-1$ ). Una rotazione di  $D$  posizioni può essere definita concettualmente come l'effetto di  $D$  rotazioni elementari. Si realizzi una funzione  $C$  di rotazione a destra o a sinistra di  $D$  posizioni, dove  $D$  è un intero (positivo per rotazione a destra, negativo a sinistra).

Ad esempio, che una rotazione, con  $N=10$  e  $D=3$ , sul vettore: 5,4,10,-1,7,20,11,-3,6,34, produrrebbe il vettore: -3,6,34,5,4,10,-1,7,20,11.

E' vietato realizzare la rotazione di  $D$  posizioni come la sequenza di  $D$  rotazioni da 1 posizione ciascuna, che avrebbe costo  $O(N*D)$ . Si richiede invece un algoritmo  $O(N)$ , che utilizzi un vettore ausiliario di  $D$  posizioni, allocato dinamicamente, in cui depositare temporaneamente  $D$  elementi. Dopo aver spostato gli altri  $N-D$  elementi, ciascuno nella sua casella destinazione, anche i  $D$  elementi salvati vanno collocati nella loro destinazione finale.

### 2. (4 punti)

Una matrice si dice "sparsa" quando la maggior parte delle caselle contiene un valore nullo. Nel caso di una matrice sparsa, può risultare conveniente l'utilizzo di una struttura dati che allochi dinamicamente memoria per le sole caselle non nulle.

Si realizzi in C, per una matrice sparsa di numeri (tipo float), gestita come ADT di prima categoria, la funzione di prototipo

```
MATRwrite (matr_t *M, int r, int c, float val);
```

che inserisca alla riga  $r$  e alla colonna  $c$  il valore  $val$ . Il tipo `matr_t` è una struct che, oltre a contenere, come interi, le due dimensioni della matrice ( $NR$ , numero di righe, e  $NC$ , numero di colonne) punta a un vettore (di dimensione  $NR$ ) di puntatori a righe, ognuna realizzata mediante una lista, contenente gli elementi non nulli della riga corrispondente). Si definisca `matr_t`, si definisca poi il tipo struct usato per le liste e si scriva infine la funzione `MATRwrite`.

### 3. (6 punti)

Si scriva in C una funzione ricorsiva in grado di generare tutte le stringhe di lunghezza  $N$  costituite dalle 5 vocali maiuscole 'A', 'E', 'I', 'O', 'U' con i seguenti vincoli:

- il valore di  $N$  sia un parametro della funzione e sia  $N \geq 5$
- nella stringa generata ogni vocale compaia almeno una volta.

All' $i$ -esimo livello di ricorsione:

- la funzione gestisca l' $i$ -esima cella della stringa della soluzione
- le scelte siano le 5 vocali maiuscole.

#### 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 giovedì 27/02/2014, alle ore 23:59, via e-mail all'indirizzo: [danilo.vendramineto@polito.it](mailto: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.

**PER I LAUREANDI:** per il calendario di invio relazioni ed esami orali vedere avviso sulla pagina del corso 02MNO AP sul Portale.