

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

Appello del 02/02/2015 - 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  $j$ :

3-9, 4-3, 2-5, 6-2, 3-8, 5-6, 0-10, 3-5, 8-9, 10-3

si applichi un algoritmo di on-line connectivity con weighted 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 il seguente vettore di interi mediante Shellsort con la sequenza di Knuth:

5 4 10 7 6 4 0 1 6 5 0 2 7 5 0 3 0 4 9

Si indichino i passaggi principali.

### 3. (2 punti)

Sia data una coda a priorità inizialmente vuota implementata mediante uno heap. Sia data la sequenza di interi e carattere \*:

20 32 19 51 \* \* 28 \* 74 9 81 \* \* 17 \* 41 33 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. Al termine si cambi la priorità di 41 in 11 e si disegni la configurazione risultante della coda a priorità.

### 4. (1 punto)

Si esprima in notazione prefissa e postfissa la seguente espressione aritmetica mediante visita dell'albero binario corrispondente:

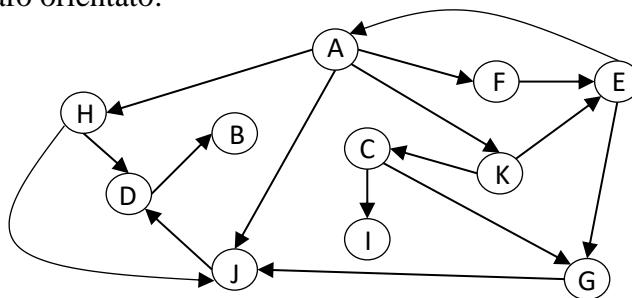
$((A + B) * (C * (D - E))) / ((E - F) + G * H)$

### 5. (2 punti)

Sia data la sequenza di chiavi FEB<sub>1</sub>B<sub>2</sub>RAIO, dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto (A=1, ..., Z=26) e da un eventuale pedice. Si riporti la struttura di una tabella di hash di dimensione 19, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza indicata. Si supponga di utilizzare l'open addressing con quadratic probing. Si selezionino opportuni valori per  $c_1$  e  $c_2$ .

### 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 02/02/2015 - Prova di programmazione (18 punti)

**Formulazione del problema:** uno studente X impegnato a preparare gli esami della sessione invernale si è scordato di fare i regali di Natale ai suoi amici. Alcuni di questi amici sono a loro volta amici tra loro e la loro relazione di amicizia è descritta mediante un grafo non orientato. Ogni amico dello studente X deve ricevere da X uno ed un solo regalo, più amici di X possono ricevere da X regali dello stesso tipo alla condizione di non essere amici tra di loro.

### Dati in ingresso:

- le relazioni di amicizia tra gli amici di X sono memorizzate in un primo file di testo, dove per ogni riga compare una coppia di nomi di 2 amici di X che sono a loro volta amici tra di loro. Il numero di righe del file non è noto a priori, né sovrapstimabile
- lo studente X ha prodotto manualmente una soluzione scritta su di un secondo file di testo. Nella prima riga si trova il numero N di tipologie di regali che X ha determinato. Di seguito compaiono tanti gruppi di righe quante sono le N tipologie di regali. Ciascun gruppo consta di una prima riga con un intero  $k_i$  ( $1 \leq k_i \leq \text{numero\_amici}$ ) che indica il numero di amici che riceveranno un regalo appartenente alla tipologia corrente  $i$  ( $0 \leq i < N$ ), seguito nelle righe  $k_i$  successive dai nomi degli amici uno per riga.

### Vincoli:

- ogni stringa è lunga al massimo 15 caratteri
- i nomi dei file sono passati sulla riga di comando.

### Richieste:

Si scriva un programma C che:

- legga dal primo file le relazioni di amicizia tra gli amici di X, legga dal secondo file la soluzione presunta, e memorizzi le informazioni in opportune strutture dati in memoria
- verifichi che la soluzione presunta soddisfi le condizioni (ogni amico ha ricevuto uno ed un solo regalo e che due amici di X amici tra di loro non abbiano ricevuto regali dello stesso tipo). Non è richiesto di verificare che il numero tipologie di regalo sia minimo
- calcoli in modo ricorsivo il numero minimo N di tipologie di regalo tali da soddisfare le condizioni di cui sopra e generi UNA PARTIZIONE degli amici di X in N sottoinsiemi corrispondenti ognuno ad una tipologia di regalo.

### Esempio:

I file	II file
Diego Emilia	4
Claudia Emilia	1
Bartolo Claudia	Claudia
Claudia Diego	2
Bartolo Diego	Andrea
Andrea Claudia	Diego
Claudia Fabio	2
Andrea Bartolo	Bartolo
Diego Fabio	Emilia
	1
	Fabio

Il programma con questo esempio deve fornire i seguenti risultati con formato a scelta del candidato:

- punto 2: la soluzione manuale soddisfa le condizioni di cui è richiesta la verifica
- punto 3: il numero minimo di tipologie di regali è  $N=3$  e un possibile partizionamento è  $\{\text{Claudia}\}$ ,  $\{\text{Andrea, Diego}\}$ ,  $\{\text{Bartolo, Emilia, Fabio}\}$ .

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

Appello del 02/02/2015 - Prova di programmazione (12 punti)

### 1. (2 punti)

Si realizzi una funzione `C eraseDuplicate` che, data una stringa `str` ritorni la stessa stringa in cui sia mantenuta solo la prima occorrenza di tutti i caratteri che appaiono più di una volta nella stringa originale. Il prototipo della funzione è:

```
void eraseDuplicate (char *str);
```

Ad esempio, se inizialmente `str` contiene "aa;;bbbab;" la funzione deve ritornare la stringa "a;b".

### 2. (4 punti)

Sia dato un albero di grado `N`, i cui nodi sono definiti dalla seguente struttura `C`:

```
struct node {  
    int key;  
    struct node *children[N];  
};
```

Si realizzi una funzione `C`

```
void visitLevelByLevel (struct node *root, int l1, int l2);
```

che visiti l'albero da profondità `l1` a profondità `l2` e che visualizzi tutte le chiavi livello per livello, cioè prima tutte quelle a profondità `l1`, seguite da quelle a profondità `l1+1`, etc. fino a quelle a profondità `l2`. Si assuma che `l1 < l2`. Si noti che è possibile che l'albero sia visitato più di una volta per ottenere il risultato desiderato.

**Non è consentito l'uso di funzioni di libreria.**

### 3. (6 punti)

Una sigla alfanumerica di lunghezza `N` è composta selezionando per ognuna delle `N` posizioni `sigla[i]` un carattere che appartiene all'insieme `set[j]`. Ogni insieme `set[j]` è dato come stringa di caratteri alfanumerici e la sua cardinalità è al massimo 10. Un file contiene le informazioni relative alla lunghezza della sigla `N` (intero sulla prima riga) e alle stringhe che rappresentano i `set[j]` (`N` stringhe una per riga sulle `N` righe successive). Si realizzi una funzione ricorsiva in `C` che, letto il file, generi tutte le possibili sigle e le memorizzi su di un secondo file. I nomi dei file siano passati come parametri alla funzione.

Esempio : se il primo file ha il seguente contenuto :

3

A

Xy

123

occorre scrivere nel secondo file le sigle: AX1, AX2, AX3, Ay1, Ay2, Ay3.

#### **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, funzioni su `FIFO`, `LIFO`, liste, `BST`, tabelle di hash, grafi 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. I modelli delle funzioni ricorsive non sono considerati funzioni standard.
- Consegna delle relazioni (per entrambe le tipologie di prova di programmazione): entro giovedì 05/02/2015, alle ore 24:00, mediante caricamento su Portale. Le istruzioni per il caricamento sono pubblicate sul Portale nella sezione Materiale). **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.