03MNO Algoritmi e Programmazione

Appello del 04/02/2017 - Prova di teoria (12 punti)

1. (2.5 punti)

Si inseriscano in sequenza i seguenti dati, composti da una stringa e da un intero che ne rappresenta la priorità, in una coda a priorità di indici inizialmente supposta vuota:

`"si" 10  "inserisca" 15  "in" 5  "sequenza" 18  "il" 4`

Si ipotizzi di usare uno heap (priorità massima in radice, vettore di 7 celle) per la coda a priorità e che la tabella di hash di dimensione 11 per la tabella di simboli abbia il seguente contenuto:

0	1	2	3	4	5	6	7	8	9	10
	il		si	in		inserisca	sequenza			
	4		10	5		15	18			

Si disegnino lo heap e il vettore `qp` ai diversi passi dell'inserzione. Al termine si cambi la priorità di `"il"` da 4 a 20 e si disegnino i corrispondenti heap e vettore `qp`.

2. (2 punti)

Si determini mediante un algoritmo greedy un codice di Huffman ottimo per i seguenti simboli con le frequenze specificate: A: 24 B: 12 C: 10 D: 8 E: 6 M: 19 O: 14 Q: 3 T: 11 U: 7

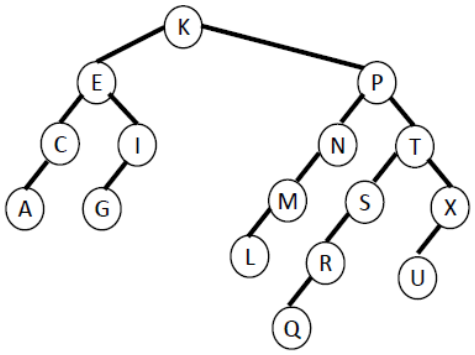
3. (2 punti)

Sia data la sequenza di chiavi intere: 11 144 267 312 98 100 45 207 13 99 181

Si riporti il contenuto di una tabella di hash di dimensione 23, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza indicata. Si usi l'open addressing con double hashing, definendo le opportune funzioni di hash.

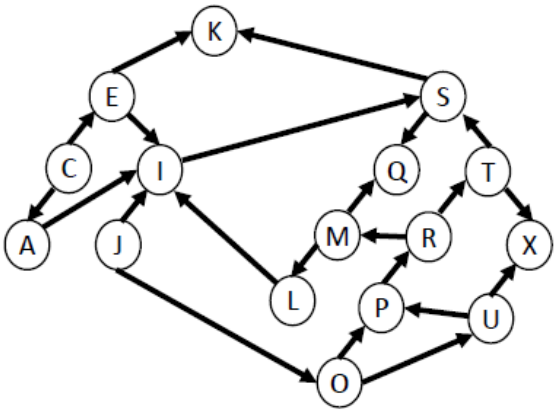
4. (2 punti)

Si partizioni il seguente BST attorno alla chiave M, evidenziando i passaggi rilevanti:



5. (2 + 1.5 punti)

Sia dato il seguente grafo orientato:



- se ne effettui una visita in profondità , considerando **S** come vertice di partenza e si etichettino i vertici con tempo di scoperta/fine elaborazione (**2 punti**)
- lo si ridisegni, etichettando ogni suo arco come T (tree), B (back), F (forward), C (cross), considerando **S** come vertice di partenza (**1.5 punti**).

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

03MNO Algoritmi e Programmazione

Appello del 04/02/2017 - Prova di programmazione (18 punti)

Un *social network* vuole focalizzare meglio le sue attività rivolte a gruppi fortemente omogenei. Per raggiungere tale scopo deve individuare questi gruppi all'interno della rete. Essa è costituita da amici tra i quali esistono relazioni di amicizia più o meno intense ed è modellata mediante un grafo non orientato e pesato G = (V, E) dove i vertici sono gli amici, gli archi le relazioni di amicizia e i pesi interi e positivi le intensità di queste relazioni.

Un gruppo omogeneo di amici corrisponde a un sottografo **completo** di G, detto **cricca** (o **clique**) in Teoria dei Grafi. Una cricca è **massimale** se e solo se non esistono altri sottografi completi di cui essa è un sottoinsieme proprio. Il grafo è memorizzato in un file mediante l'elenco dei suoi archi pesati con il seguente formato:

idV_1 idV_2 w

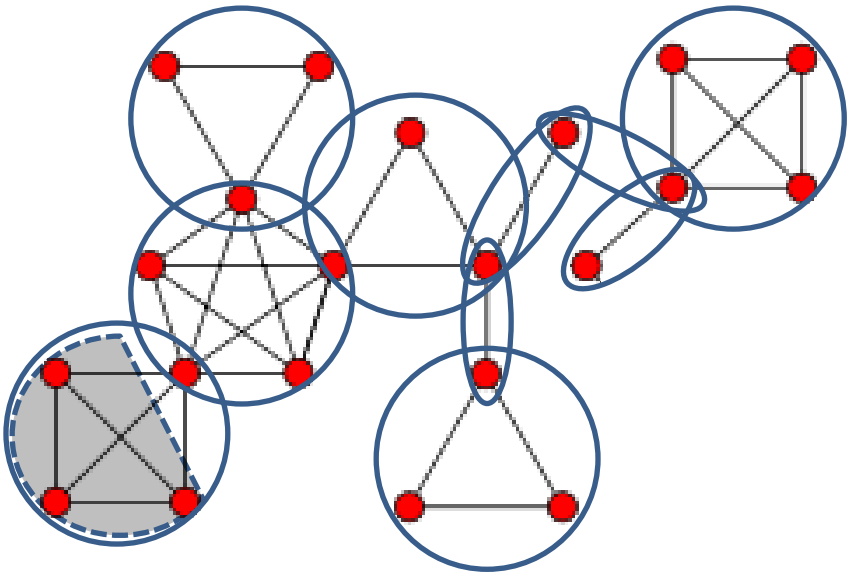
che indica che l'arco (idV_1, idV_2) ∈ E, dove idV1∈ V e idV_2∈ V e w è il peso dell'arco. Poiché il numero di vertici del grafo non è noto a priori, si suggerisce di calcolarlo tramite lettura preliminare del file e/o caricamento in una tabella di simboli. Ogni vertice è individuato mediante un identificatore alfanumerico idV di lunghezza massima uguale a 20 caratteri. Si può assumere che non esistano archi duplicati. Non è lecito assumere nessuna forma di ordinamento degli archi.

Si scriva un programma C che, ricevuti sulla riga di comando

- come primo argomento il nome del file contenente la descrizione del grafo
- come secondo argomento il nome di un file che contiene una possibile cricca (vertici uno per riga)

- verifichi se la soluzione proposta sia davvero una cricca massimale, cioè se è un sottografo completo massimale. *Si osservi che un sottografo completo non è una cricca massimale se esiste almeno un vertice ad esso esterno adiacente a tutti i suoi vertici.*
- determini tra tutte le cricche massimali quella con massimo numero di vertici
- determini tutte le cricche massimali, memorizzandole in un'opportuna struttura dati (vettore di liste). *Si noti che il massimo numero di cricche massimali è |V|*
- per ogni cricca massimale tra quelle individuate al punto precedente, identifichi il ciclo hamiltoniano (cammino semplice che tocca tutti i vertici richiudendosi su quello di partenza) a peso massimo.

Esempio:



La figura riporta cricche massimali con 2, 3, 4 e 5 vertici. Il sottografo completo di 3 vertici con sfondo grigio non è una cricca massimale, in quanto è incluso (come sottoinsieme proprio) nella cricca massimale di 4 vertici.

03MNO Algoritmi e Programmazione

Appello del 04/02/2017 - Prova di programmazione (12 punti)

1. (**2 punti**)

Data una matrice quadrata `N` x `N`, con `N` dispari, scrivere una funzione caratterizzata dal seguente prototipo:

```
void sommaCornici(int **mat, int N, int **vet);
```

che memorizzi nel vettore `vet`, allocato della dimensione opportuna dentro alla funzione, la somma degli elementi su ogni cornice della matrice.

A titolo di esempio, la seguente matrice 5x5 è caratterizzata dalla presenza di 3 cornici, evidenziate per facilità di comprensione da toni diversi (bianco, grigio chiaro, grigio scuro). Si osservi che esiste una cornice formata dal solo elemento centrale.

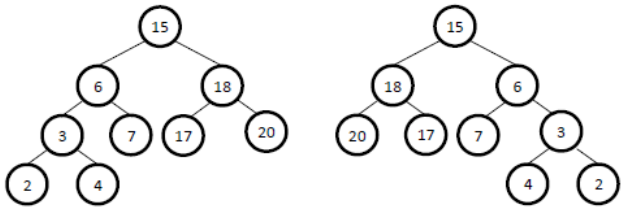
1	2	3	4	5
6	7	8	9	0
1	2	3	4	5
6	7	8	9	0
1	1	1	1	1

2. (**4 punti**)

Dato un albero binario cui si accede tramite un puntatore alla radice, si scriva una funzione C con il seguente prototipo:

```
link mirror(link root);
```

che costruisca l'albero binario "speculare", cioè quello in cui sono scambiati figlio sinistro e figlio destro, ritornando come risultato il puntatore alla sua radice, come nell'esempio seguente:



I nodi contengono interi. Non è ammesso l'uso di funzioni di libreria sugli alberi.

3. (6 punti)

Una matrice di interi `n` x `n` contiene celle bianche (1) e nere (0). Esiste un comando che, data una riga o una colonna, cambia di colore tutte le celle di quella riga o colonna.

Si scriva una funzione in grado di individuare un insieme **minimo** di comandi che, se attivati, permettano di rendere la matrice interamente bianca.

Esempio: data la matrice

0	1	0
1	0	1
0	1	0

i comandi cambia riga 0, riga 2 e colonna 1 la trasformano in una matrice di tutti 1.

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).***
- ***Se non indicato diversamente, è consentito utilizzare chiamate a funzioni <u>standard</u>, 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 devono essere allegati all'elaborato (il loro contenuto riportato nell'elaborato stesso). 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 martedì 07/02/2017, alle ore 24:00, mediante caricamento su Portale. Le istruzioni per il caricamento sono pubblicate sul Portale nella sezione Materiale). QUALORA IL CODICE CARICATO CON LA RELAZIONE NON COMPILI CORRETTAMENTE, VERRÀ APPLICATA UNA PENALIZZAZIONE. Si ricorda che la valutazione del compito viene fatta senza discussione o esame orale, <u>sulla base dell'elaborato svolto in aula</u>. Non verranno corretti i compiti di cui non sarà stata inviata la relazione nei tempi stabiliti.***