

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

Appello del 02/09/2015 - Prova di teoria (12 punti)

1. (1 punto)

Si ordini in maniera ascendente il seguente vettore di interi:

22 11 12 51 23 3 19 43 29 17 81 100 18 60 8 10

mediante insertion-sort, Si riportino i passaggi rilevanti.

2. (1 punto)

Sia dato il seguente insieme di attività, dove la i -esima attività è identificata dalla coppia tempo di inizio (s_i) e tempo di terminazione (f_i):

i	s_i	f_i	i	s_i	f_i
1	0	6	5	1	9
2	7	10	6	11	15
3	2	4	7	3	7
4	5	9	8	4	10

Si determini, mediante un algoritmo greedy, il massimo numero di attività mutuamente compatibili.

3. (2 punti)

Sia dato un albero binario con 15 nodi. Nella visita si ottengono le 3 seguenti sequenze di chiavi:

preorder	30	18	33	12	16	17	19	7	21	10	11	9	3	13	15
inorder	16	12	17	33	7	19	18	21	30	11	3	9	13	10	15
postorder	16	17	12	7	19	33	21	18	3	13	9	11	15	10	30

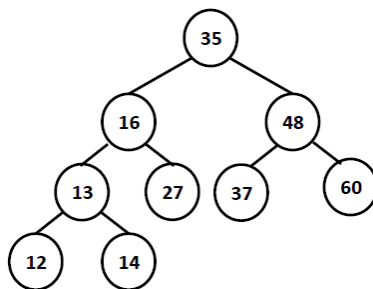
Si disegni l'albero binario di partenza.

4. (2 punti)

Sia data la sequenza di chiavi I B U X E G Z H Y, dove ciascun carattere è individuato dal suo ordine progressivo nell'alfabeto (A=1, ..., Z=26). Si riporti la struttura di una tabella di hash di dimensione 19, 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.

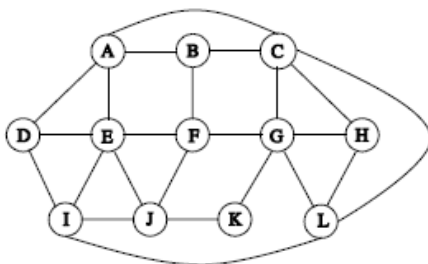
5. (2 punti)

Si partizioni il seguente BST intorno alla chiave mediana:



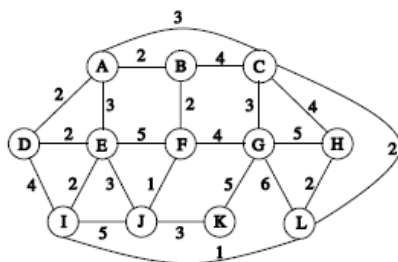
6. (2 punti)

Si visiti in profondità il seguente grafo non orientato, 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. Qualora necessario, si considerino i nodi in ordine alfabetico.



7. (2 punti)

Sia dato il seguente grafo non orientato pesato:



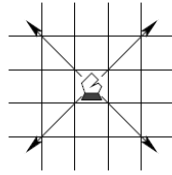
se ne determini un minimum spanning tree applicando l'algoritmo di Prim a partire dal vertice A, disegnando l'albero e ritornando come risultato il valore del peso minimo. Si esplicitino i passi intermedi.

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

Appello del 02/09/2015 - Prova di programmazione (12 punti)

1. (2 punti)

Nel gioco degli scacchi l'alfiere si muove di un numero di caselle arbitrario lungo la diagonale e la diagonale inversa come rappresentato in figura:



Negli scacchi inoltre ogni pezzo assume idealmente un certo valore, ad esempio 1 per il pedone, 3 per il cavallo, etc. Con questo principio le caselle vuote assumono valore uguale a 0. Si realizzi un funzione in C che data una matrice di interi di dimensione $N \times N$ nota e contenente i pezzi degli scacchi con i loro pesi nonché celle vuote, calcoli e visualizzi le coordinate delle casella vuota della scacchiera nella quale occorrerebbe disporre un alfiere in modo che la somma dei valori dei pezzi disposti sulla stessa diagonale e diagonale inversa sia massima. Sia dato il seguente esempio di scacchiera 4×4 :

	1	2	3	4
1	0	3	4	0
2	1	0	6	6
3	1	3	9	0
4	0	0	3	1

In questo caso occorre posizionare l'alfiere nella cella di riga 4 e colonna 2 (con somma 16).

2. (4 punti)

Siano dati i puntatori $t1$ e $t2$ a 2 alberi binari. Si scriva una funzione C

```
int TREEisomorph(struct node *t1, struct node *t2);
```

che verifichi se i 2 alberi sono o meno isomorfi. Per isomorfi si intendono alberi strutturalmente identici, cioè che contengono gli stessi valori disposti allo stesso modo. Si ipotizzi di avere a disposizione una funzione di comparazione tra chiavi

```
int KEYcmp(Key k1, Key k2);
```

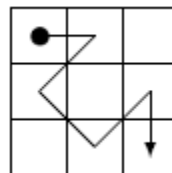
3. (6 punti)

Una mappa $N \times M$ contiene in ogni cella un valore intero. Nella mappa la posizione di partenza è sempre l'angolo in alto a sinistra, mentre quella di arrivo è sempre l'angolo in basso a destra. Sono ammessi spostamenti in tutte le celle adiacenti a quella corrente, inclusi quelli in diagonale, purché non si ripassi sulla medesima casella.

Si scriva una funzione C che, ricevuto come parametro la matrice contenente la mappa, individui un cammino che permetta di spostarsi dalla partenza all'arrivo massimizzando la somma dei valori delle celle attraversate. A parità di valore complessivo, inoltre, l'algoritmo deve prediligere soluzioni che richiedano il minor numero possibile di passi.

Esempio: per la mappa della figura a sinistra, una possibile soluzione è quella della figura a destra con peso 24:

1	2	-3
9	-9	7
0	1	4

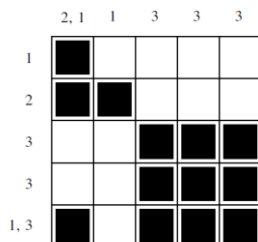


02MNO Algoritmi e Programmazione 01JKE APA I / 01JKF APA II

Appello del 02/09/2015 - Prova di programmazione (18 punti)

I **Nonogram** o **Paint by Numbers** (dipingere con i numeri) o **griddlers**, sono dei rompicapi logici grafici in cui le celle di una griglia $N \times M$ devono essere colorate o lasciate in bianco in base a dei numeri che compaiono a lato delle righe e delle colonne della griglia. In questo tipo di rompicapo, ogni numero indica quante celle consecutive devono essere riempite in una riga o in una colonna. Per esempio, un'etichetta di riga/colonna del tipo "4 8 3" significa che su quella riga/colonna ci sono 3 insiemi di 4, 8 e 3 celle nere rispettivamente da riempire in questo ordine, con almeno una cella bianca tra gruppi successivi di celle nere. Il gioco consiste nell'individuare una configurazione di celle nere congruente con l'etichettatura delle righe e delle colonne.

Esempio di versione originale del gioco dove il numero di gruppi di celle contigue colorate non è dato esplicitamente, ma si ricava.



Un primo file di testo il cui nome è passato sulla riga di comando contiene:

- nella prima riga del file il numero di righe N della griglia
- per ogni riga della griglia un gruppo di informazioni (una per riga del file):
 - numero di gruppi ni di celle nere contigue per quella riga della griglia (semplificazione rispetto alla versione originale del gioco)
 - ni interi che rappresentano la lunghezza di ciascuno di questi insiemi
- segue su una riga del file il numero di colonne M della scacchiera
- per ogni colonna della griglia un gruppo di informazioni (una per riga del file):
 - numero di gruppi ni di celle nere contigue per quella colonna della griglia (semplificazione rispetto alla versione originale del gioco)
 - ni interi che rappresentano la lunghezza di ciascuno di questi insiemi.

Per l'esempio di cui sopra il file conterrebbe:

```
5
1 1
1 2
1 3
1 3
2 1 3
5
2 2 1
1 1
1 3
1 3
1 3
```

Un secondo file di testo il cui nome è passato sulla riga di comando contiene una possibile soluzione da verificare nella forma di N righe ciascuna contenente M interi che valgono 0 se la cella della griglia a quella riga e quella colonna è vuota, 1 se è piena.

Si scriva un programma C che:

1. legga i vincoli dal primo file di testo e li memorizzi in un'opportuna struttura dati in memoria
2. verifichi che la soluzione calcolata a mano e letta dal secondo file soddisfi le condizioni del nonogram
3. calcoli in modo ricorsivo una soluzione del gioco e la scriva su un terzo file di testo il cui nome è passato sulla riga di comandi. Il formato di questo terzo file sia lo stesso del secondo file.

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 lunedì 07/09/2015, 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.