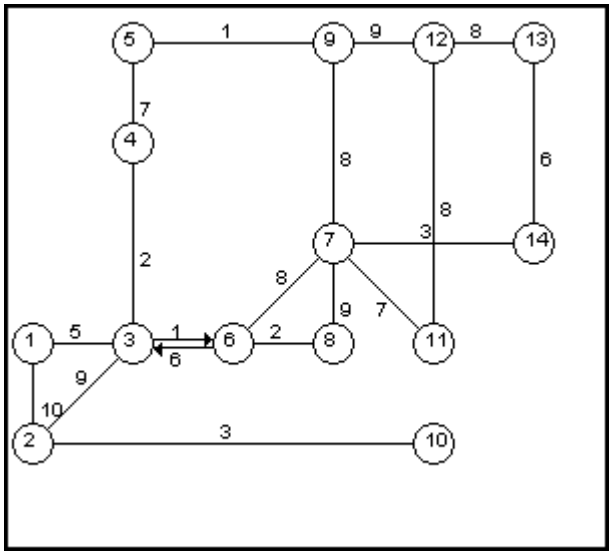
Algoritmi e Strutture Dati II
4 ottobre 2007

Svolgere esattamente 3 dei seguenti esercizi:

1) Si consideri il seguente grafo orientato e pesato:



in cui un arco non orientato di peso *x* sta ad indicare due archi orientati aventi lo stesso peso *x*.
Si supponga che sia stato preventivamente calcolato il vettore d[] delle distanze dal nodo 1 ad ogni altro nodo, ottenendo:
d[1]=0, d[2]= 10, d[3]= 5, d[4]=7, d[5]= 14, d[6]=6, d[7]=14, d[8]= 8,
d[9]=15, d[10]=13, d[11]=21, d[12]= 24, d[13]=23, d[14]=17.
Calcolare il cammino minimo dal nodo 1 al nodo 13.

2) Si considerino le due sequenze X= < A, C, D, A, A, D, A, B > e Y = < D, C, A, A, C, D, B, A >.
Trovare la sottosequenza più lunga comune alle due stringhe X ed Y, simulando l'esecuzione dell'algoritmo noto.

3) Sia dato l'alfabeto Σ={0, 1, 2, 3, 4}. Simulare l'esecuzione dell'algoritmo di Rabin-Karp (senza utilizzare la riduzione *mod p*, al fine di facilitare lo svolgimento dell'esercizio) per cercare tutte le occorrenze della stringa 144 all'interno della stringa 01312144.

4) Siano date le matrici M_1, M_2, M_3, M_4 di dimensione rispettivamente 10x5, 5x8, 8x5, 5x5.
Trovare la disposizione ottimale delle parentesi che minimizza il costo del calcolo del prodotto $M_1\ M_2\ M_3\ M_4$ ed il numero di moltiplicazioni elementari richieste utilizzando tale disposizione.

5) Sia data la partizione {{1}{2}{3}{4}{5}{6}{7}{8}{9}{10}}. Mostrare come si modifica la foresta di alberi che la rappresenta in seguito all'applicazione delle funzioni:

union(find(1), find(2)) union(find(1),find(3)) union(find(1),find(4))
union(find(1),find(5)) union(find(6),find(7)) union(find(7),find(8))
union(find(8),find(9)) union(find(9),find(3))

utilizzando le euristiche note.

Rispondere ad esattamente 3 delle seguenti domande:

1. Nel contesto della rappresentazione di insiemi disgiunti, illustrare e discutere le euristiche utilizzate per migliorare l'efficenza delle operazioni di Union e Find.
2. Nel contesto del problema della Ricerca dei Cammini Minimi, illustrare e discutere l'algoritmo per costruire i cammini minimi, qualora siano note le distanze dalla sorgente x ad ogni altro vertice.
3. Dimostrare il seguente Teorema: *Sia G=(V, E) un grafo non orientato, connesso e pesato. Siano, inoltre, m ed n rispettivamente il numero di archi ed il numero dei vertici del grafo. La complessità dell'algoritmo di Kruskal nel caso peggiore è O(m log n).*
4. Enunciare e dimostrare il Teorema Fondamentale utilizzato per calcolare la più lunga sottosequenza comune (Sottostruttura ottima di una LCS).
5. Illustrare e discutere l'algoritmo (basato sulla programmazione dinamica) utilizzato per ottenere la disposizione delle parentesi che minimizza il numero di moltiplicazioni elementari nel prodotto di n matrici assegnate.