# Algoritmi e Strutture Dati II
# 15/12/08

**Svolgere esattamente tre dei seguenti esercizi:**

a) Sia dato il grafo G=(V, E), dove V={1, 2, 3, 4, 5, 6, 7} ed E={(1,2), (1,3), (1,4), (4,1), (4,6), (3,5), (2,3), (5,6), (6,7)}. Rappresentare il grafo G utilizzando liste di adiacenza. Simulare l'esecuzione dell'algoritmo di visita in profondità a partire dal vertice 1, mostrando l'evoluzione del contenuto delle strutture dati utilizzate (ovvero, mostrando ad ogni passo cosa contengono tutte le variabili utilizzate).

b) 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.

c) Sia dato il grafo non orientato e pesato G=(V, E), dove V={1, 2, 3, 4, 5, 6, 7} ed E={[1,2], [1,3], [4,2], [4,1], [3,5], [2,3], [5,6], [6,7]}, in cui gli archi {[1,2], [1,3]} hanno peso 1, gli archi {[4,2], [4,1]} hanno peso 2, mentre gli altri hanno peso 3. Trovare il minimo albero ricoprente simulando l'esecuzione dell'algoritmo di Kruskal.

d) Siano date le matrici M1, M2, M3, M4 di dimensione rispettivamente 2x3, 3x8, 8x4, 4x2. Trovare la disposizione ottimale delle parentesi che minimizza il costo del calcolo del prodotto M1 M2 M3 M4 ed il numero di moltiplicazioni elementari richieste utilizzando tale disposizione.

**Rispondere ad esattamente due delle seguenti domande:**

a) Illustrare e discutere l'algoritmo noto per costruire la chiusura transitiva di un grafo orientato.

c) Enunciare e dimostrare il Teorema Fondamentale utilizzato per calcolare la più lunga sottosequenza comune (Sottostruttura ottima di una LCS).

b) 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).*