

Relazione

Strutture dati

Il grafo letto da file è in formato standard quindi utilizzo la funzione standard di lettura di un grafo (GraphLoad) che memorizza la matrice delle adiacenze, il numero di vertici e di archi, e crea (GaphEdge) un apposito vettore di strutture di archi in cui i vertici sono rappresentati da indici interi e il peso è un numero intero non negativo. Il grafo è realizzato come ADT di prima classe quindi il client vede solo il puntatore "opaco" alla struttura wrapper che contiene anche la tabella di simboli. In questo caso la tabella di simboli è quindi interna al grafo ed è un semplice vettore che associa a una stringa (il vertice) un indice, quello che viene usato nella struttura Edge.

Punto 1 e 2

Per individuare un arco che se rimosso rende il grafo un DAG è sufficiente una visita in profondità DFS durante la quale utilizzando tempi di scoperta e di fine elaborazione si individuano gli archi Back, quelli che connettono un vertice a un antenato nell'albero generato dalla visita in profondità. Questi sono quelli che creano un ciclo. Per trovare tutti gli insiemi di archi di cardinalità minima si utilizza un powerset con combinazioni semplici per ottenere gli insiemi di 1,2,3... archi che se rimossi rendono il grafo un DAG. Utilizzando le combinazioni siamo sicuri che il primo insieme di archi trovato sarà quello a cardinalità minima. Si procede quindi a rimuovere un arco alla volta e, con una DFS, verificare se ora il grafo è un DAG. In caso positivo gli archi, che vengono indicizzati, sono salvati in un vettore e ne viene calcolato il peso totale. Con un semplice confronto e l'utilizzo di una variabile (max) passata "by pointer" si trova quale insieme abbia peso massimo. Gli archi da eliminare vengono stampati e successivamente viene creato il DAG (dagcreate) eliminando quegli archi e aggiornando il vettore di strutture di archi.

Punto 3

Una volta ottenuto il DAG si procede alla sua riorganizzazione e stampa in ordine topologico. Questo è possibile grazie a un'ulteriore DFS durante la quale vengono salvati i tempi di fine elaborazione in un vettore di interi (st). A questo punto l'ordinamento topologico, per cui tutti gli archi vanno da destra a sinistra nell'albero orizzontale della visita in profondità, è ricavato scandendo in ordine inverso st.

Il problema dei cammini massimi per DAG pesati prevede il calcolo delle distanze massime da ogni vertice a quelli da esso raggiungibili. I vertici vengono presi in ordine topologico e utilizzando un vettore per la massima distanza di ogni vertice dal vertice di partenza e uno per ricostruire i cammini massimi, si applica la Relaxation inversa per ogni arco che viene considerato. A questo punto si procede alla stampa dei cammini massimi da ogni vertice indicando con -1 se il vertice non è raggiungibile da quello di partenza e indicando anche quale è stato l'ultimo vertice del cammino prima di raggiungere il vertice di destinazione.