Relazione Elaborato 1

Scopo

Ricavare il centro di un grafo. Per farlo dobbiamo trovare i vertici la cui massima distanza dagli altri nodi del grafo è minima questo corrisponde a trovare l'insieme dei vertici con eccentricità minore. L'eccentricità di un grafo è così definita
$\epsilon(i) = max_{i \in V}(min_{i,j \in V}(d(i,j)))$. Per calcolarla abbiamo bisogno quindi di trovare i cammini minimi da ogni nodo verso tutti gl'altri. Per costo di un cammino si intende la somma dei pesi degli archi che lo definiscono. Dovremmo trovare tutti i cammini da un nodo verso tutti gl'altri con il costo minore praticamente il nostro algoritmo dovrà saper valutare quale percorso è il migliore, il meno costoso, per arrivare dal nodo A al nodo B, ciò potrebbe voler dire che il cammino in questione sia composto da nodi e archi intermedi che se sommati danno un peso inferiore a quello dell'arco da A a B.

Funzionamento dell'algoritmo e confronti

L'algoritmo da noi utilizzato è Floyd-Warshall seguito da dei controlli. Grazie a questa scelta abbiamo facilmente potuto individuare tutti i cammini minimi. Il funzionamento deriva dall'utilizzo della programmazione dinamica che ci permette di memorizzare dei risultati intermedi senza doverli ricalcolare ogni volta. Nello specifico vengono eseguiti tre cicli for annidati che confrontano se è più conveniente andare da A a B direttamente o passando per un predecessore di B. Tutto questo viene fatto con complessità computazionale O(V^3). La struttura dati su cui esso si basa ovvero una matrice delle distanze è esattamente il tipo di dato che avevamo in input a seguito della lettura del file. Le alternative erano Dijkstra e Bellman-Ford entrambi permettono di calcolare cammini minimi ma a sorgente singola ovvero tra un nodo e l'altro. Avremmo dovuto ripetere questi algoritmi n volte almeno per ottenere lo stesso risultato di Floyd-Warshall.

Progetto in generale

All'interno del Main ci occupiamo di aprire il file in modalità lettura. Successivamente la funzione allocazione alloca una matrice n per n dove n è il numero dei nodi e la inizializza a distanza infinita con degli INT_MAX. Successivamente ci occupiamo di leggere tutti i valori del file inserendoli in modo tale da avere nelle righe il nodo di partenza, nelle colonne quello di arrivo e nella cella di incrocio la distanza tra uno e l'altro, verifichiamo di leggere la distanza minore tra due stessi nodi. Eseguiamo Floyd-Warshall. Chiamiamo ora una funzione che calcola l'eccentricità questa si occupa di prendere il massimo valore diverso dalla distanza minima di ogni nodo verso tutti gl'altri. Tramite controllo verifichiamo anche che qualora il nodo in questione non abbia archi uscenti , intera riga di INT_MAX, inseriremo INT_MAX come distanza massima di quel nodo da tutti gl'altri. Successivamente cerchiamo il minimo tra i massimi ovvero l'eccentricità. Infine scriviamo su file i nodi che hanno l'eccentricità trovata.

Tempi di esecuzione

Nella pagina che segue è rappresentata la tabella con le tempistiche di esecuzione del programma. Sono esclusi gli intervalli di tempo per il caricamento e la lettura del file. Quando andiamo a compilare, prima in modalità Debug e poi in modalità Release, si notano chiaramente le differenze di tempo che ci sono, man mano che si aumenta il numero di nodi. Questo è stato il motivo che ci ha spinto a inserire entrambi le modalità nella tabella. Per avere un risultato più attendibile, per ogni valore possibile son stati effettuati 4 test e in seguito ne è stata ricavata la media aritmetica. Il valore che si legge in tabella è proprio il risultato della media per quello specifico caso. Va specificato che il programma è stato testato su un processore Intel Pentium N4200 1,1 GHz.

file.dat	debug	release
g100dense	≃ 0,009 sec	≃ 0,009 sec
g100sparse	≃ 0,011 sec	≃ 0,011 sec
g200dense	≃ 0,056 sec	≃ 0,049 sec
g200sparse	≃ 0,070 sec	≃ 0,062 sec
g500dense	≃ 0,713 sec	≃ 0,368 sec
g500sparse	≃ 0,858 sec	≃ 0,397 sec
g1000dense	≃ 4,421 sec	≃ 2,391sec
g1000sparse	≃ 4,536 sec	≃ 2,584 sec

Conclusione

Ciò che si evince dai test effettuati e dai tempi di esecuzione, è che la scelta di creare questo programma con l'algoritmo di Floyd-Warshall e le strutture dati precedentemente viste (dopo aver studiato attentamente anche le altre possibilità), si è rivelata una scelta giusta, in quanto ci porta a un'ottima soluzione del problema iniziale.