

Algoritmi e Strutture Dati

Progetto di Laboratorio Esercizio 3/4

Anno Accademico: 2022/2023

AbdelHamid Badrane[matricola: 902154]

abdelhamid.badrane@edu.unito.it

Marika Torrente[matricola: 916315]

marika.torrente@edu.unito.it

Sommario

1.Esercizio 4

- 1.1 Introduzione
- 1.2 Struttura dati Grafo
- 1.3 Algoritmo Prim
- 1.4 Conclusioni

1.1 Introduzione

L'esercizio 4 prevede l'implementazione di una libreria che realizza la struttura dati Grafo che sfrutta l'algoritmo di Prim che a sua volta utilizza la PriorityQueue min-heap realizzata nell'esercizio(3) precedente, per identificare l'mst ovvero la minima foresta ricoprente a partire da un file csv passato come argomento.

1.2 Struttura dati Grafo

Per quanto riguarda la libreria per la creazione di grafi è costruita in modo da permettere la creazione di archi diretti, indiretti, etichettati e non etichettati.

La struttura dati grafo è quindi implementata grazie ad una struttura di supporto Arco(Edge) basata su nodo di inizio, nodo di fine e etichetta tra i due(l'etichetta indica il peso dell'arco). Un arco diretto è un arco che collega due nodi tra di loro con una direzione, parte dal primo nodo e va verso il secondo. Un arco indiretto, invece, è un arco composto da una coppia di archi di andata e ritorno con lo stesso peso.

L'etichettatura permette di inserire i pesi dei vari archi in modo da poter calcolare un mst, nel caso non siano etichettati l'mst non avrà un unico risultato definito, poiché composto da molti alberi con copertura minima con lo stesso peso totale.

L'implementazione della libreria oltre ad offrire i metodi per l'interazione con la struttura dati, mantiene anche le informazioni riguardanti i nodi che vengono salvati come chiavi, mentre il valore viene rappresentato da una collection di Set della classe Edge, quest'ultima entità rappresenta gli archi collegati al vertice chiave e mantiene oltre ai vertici collegati anche il peso di caduno.

1.3 Algoritmo Prim

L'algoritmo di Prim per determinare la foresta minima coprente all'interno di un grafo pesato. L'algoritmo si appoggia per la sua esecuzione alle seguenti strutture dati:

- una mappa contenente le distanze minime con chiave il nodo e valore la distanza minima.
- una mappa contenente i nodi e i propri predecessori a costo minimo.
- una mappa utilizzata per tenere traccia dell'arco più vicino per ogni nodo in modo da andare a formare l'mst, questa hashmap usa come chiave i nodi e come elemento tiene traccia degli archi per raggiungere tutti i nodi all'interno del grafo partendo da source; gli archi contengono il vertice predecessore e il peso dal vertice sorgente
- una coda di priorità minima implementata grazie all'utilizzo della libreria PriorityQueue sviluppata per l'esercizio precedente che ha il compito di tenere traccia dell'arco al prossimo vertice con peso minore.

1.3 Conclusioni

L'esercizio permette di trovare come richiesto la minima foresta ricoprente, e quindi anche tutti gli archi che collegano i vertici tra le due città in modo da poter raggiungere tutte le destinazioni all'interno del grafo.

Tramite l'utilizzo di grafo etichettato con archi indiretti troviamo un totale di 18637 archi con peso complessivo di 89.939,91258557298 km