

Esercitazione 3

Si implementi un'applicazione web che, estendendo l'applicazione sviluppata per la seconda esercitazione, permetta di visualizzare il percorso più breve tra due punti scelti dalla mappa della città.

Si proceda come di seguito descritto:

1. Si implementi un'applicazione Java standalone che crei, a partire dalla tabella BusStop presente nella base dati della scorsa esercitazione, la versione "*geografica*" delle fermate (vedi slide 09-spatial.pptx).
2. Si crei un container Docker con MongoDB (a partire dall'immagine ufficiale presente sul DockerHub) e lo si configuri affinché sia raggiungibile dall'host.
3. Si implementi un'applicazione Java standalone che calcoli, per ogni fermata presente nella base dati, il percorso più breve verso tutte le altre fermate:
 - a. Implementare le classi necessarie per costruire un grafo direzionale, i cui nodi rappresentino le fermate e gli archi il costo tra una nodo e il successivo (per il problema da risolvere si consiglia di rappresentare il grafo con una lista di adiacenze);
 - b. Costruire il grafo inserendo ogni fermata e collegando ad esse, sia le fermate direttamente raggiungibili con una bus, sia quelle non collegate da una linea ma distanti non più di 250m, raggiungibili quindi a piedi (utilizzare Postgis per interrogare il database e trovare le fermate presenti in un dato raggio).
 - i. Per i costi da assegnare agli archi, si assuma per semplicità un costo dipendente dalla sola distanza tra i punti senza tenere in considerazione gli orari dei pullman alle fermate;
 - ii. Si scelga una ragionevole costante moltiplicativa per il costo degli archi percorsi a piedi.
 - c. Implementare l'algoritmo di Dijkstra (o una sua variante) per calcolare il percorso a costo minimo tra una fermata e tutte le altre;
 - d. Salvare ogni percorso minimo trovato all'interno di una collezione di MongoDB. La struttura dei documenti dovrà rispecchiare la struttura della classe MinPath.java in allegato.
4. Estendere l'applicazione web della scorsa esercitazione aggiungendo una pagina che presenti una mappa della città di Torino su cui l'utente potrà, cliccando sulla mappa, scegliere due punti, partenza e destinazione. Interrogare Postgis per trovare le fermate più vicine ai punti di partenza e destinazione (in un raggio massimo di 250m); interrogare successivamente MongoDB per trovare tutti i percorsi aventi come partenza e destinazione le fermate trovate; restituire il percorso più breve e visualizzarlo all'utente sulla mappa insieme alle informazioni relative alle linee coinvolte nel percorso.
5. Facoltativo in alternativa al punto 4, reimplementare l'intera applicazione web utilizzando Spring invece di JavaEE.