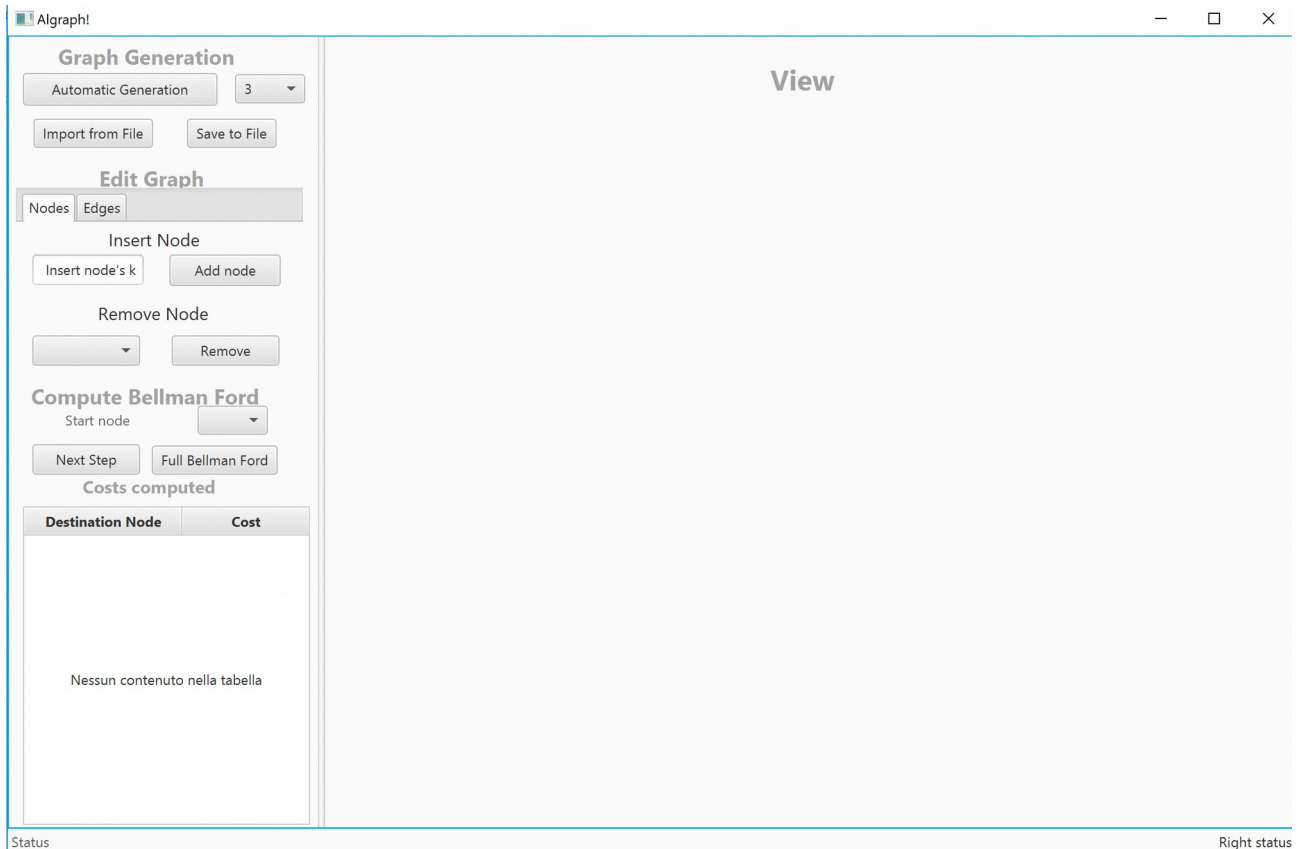


Algraph

Gruppo: Codicè-Ventura

Algoritmo: Bellman-Ford



[finestra iniziale dell' applicazione: si noti la struttura bipartita con a sinistra le parti relative alle modifiche del grafo e alla computazione dell' algoritmo e a destra il pannello view ove verrà disegnato il grafo]

Struttura

Per sviluppare il software Algraph abbiamo deciso di strutturare il progetto in due diverse macrocomponenti che interagiscono l'un l'altra: l'una, contenuta nella cartella Graph, contiene le strutture dati necessarie (Interfaccia della classe Graph e classi Graph e Node) mentre l'altra, contenuta nella cartella Algraph, contiene i file riguardanti la grafica. Nello specifico abbiamo deciso di includere nella classe Graph tutte le funzioni necessarie all'implementazione di un grafo diretto/pesato e l'algoritmo di Bellman-Ford, la classe viene poi utilizzata estensivamente

dalle componenti "grafiche" del progetto quali la classe GraphUtils, che disegna il grafo in un oggetto Pane (Java FX) e la classe Controller, che si occupa di generare il grafo (randomicamente o tramite input da file), di gestire le modifiche da parte dell' utente (aggiunta/rimozione archi e nodi) e di animare, colorando i nodi, l' esecuzione di Bellman-Ford.

Per la costruzione dell' interfaccia grafica abbiamo usato SceneBuilder il quale genera un file (chiamato Algraph_template.fxml) che viene poi usato dalla classe Controller per dare semantica e funzionalita' ai vari oggetti grafici creati con SceneBuilder.

Come strutture dati per l' implementazione del grafo abbiamo usato una `HashMap<Node, HashMap<Node,Integer>>`

che associa ad ogni nodo un'altra HashMap che a sua volta contiene un' associazione nodo di destinazione - costo.

Per disegnare il grafo invece ci siamo serviti degli oggetti Circle

(<https://docs.oracle.com/javase/8/javafx/api/javafx/scene/shape/Circle.html>) per i nodi e degli oggetti Line

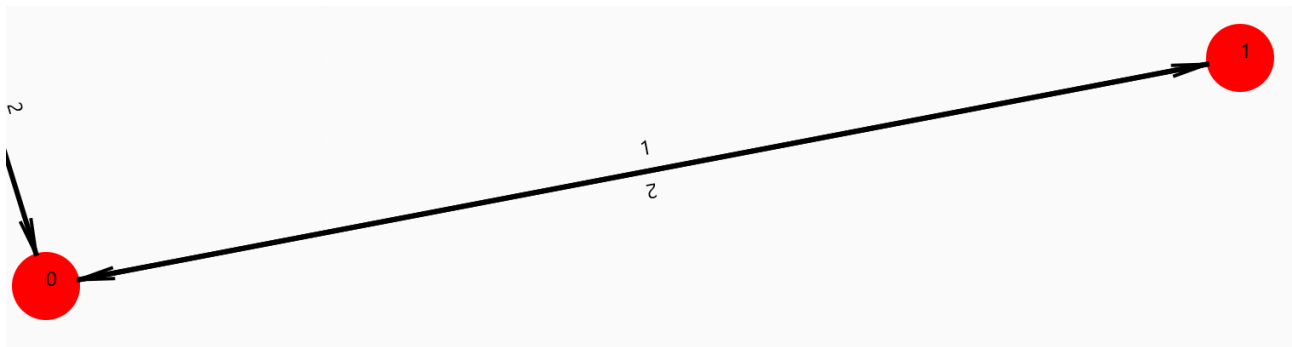
(<https://docs.oracle.com/javase/8/javafx/api/javafx/scene/shape/Line.html>) per gli archi.

Questi Circles vengono poi stampati, deterministicamente a seconda del loro numero, come in una matrice ma spostati di un offset per far meglio vedere gli archi.

Pertanto ogni grafo con lo stesso numero di nodi sarà disegnato in modo uguale e non appena se ne aggiungerà (o rimuoverà) uno l' intera struttura del grafo cambierà.

Gli archi invece vengono costruiti in modo molto più ovvio creando una linea dal nodo di partenza al nodo di arrivo e utilizzando l' arcotangente fra i due nodi per stabilirne l' angolo e costruire successivamente le linee che andranno a costituire la freccia.

Due archi opposti dunque si troveranno sulla stessa Linea ma saranno facilmente riconoscibili ed identificabili (a parte la cuspidè ad entrambe le estremità) dal fatto che il peso relativo sarà sempre stampato sopra l' arco che va da sinistra a destra così che se vi saranno due archi opposti i pesi appariranno rotati di 180 gradi l' un l' altro, rendendo immediato il riconoscimento dell' arco in esame.

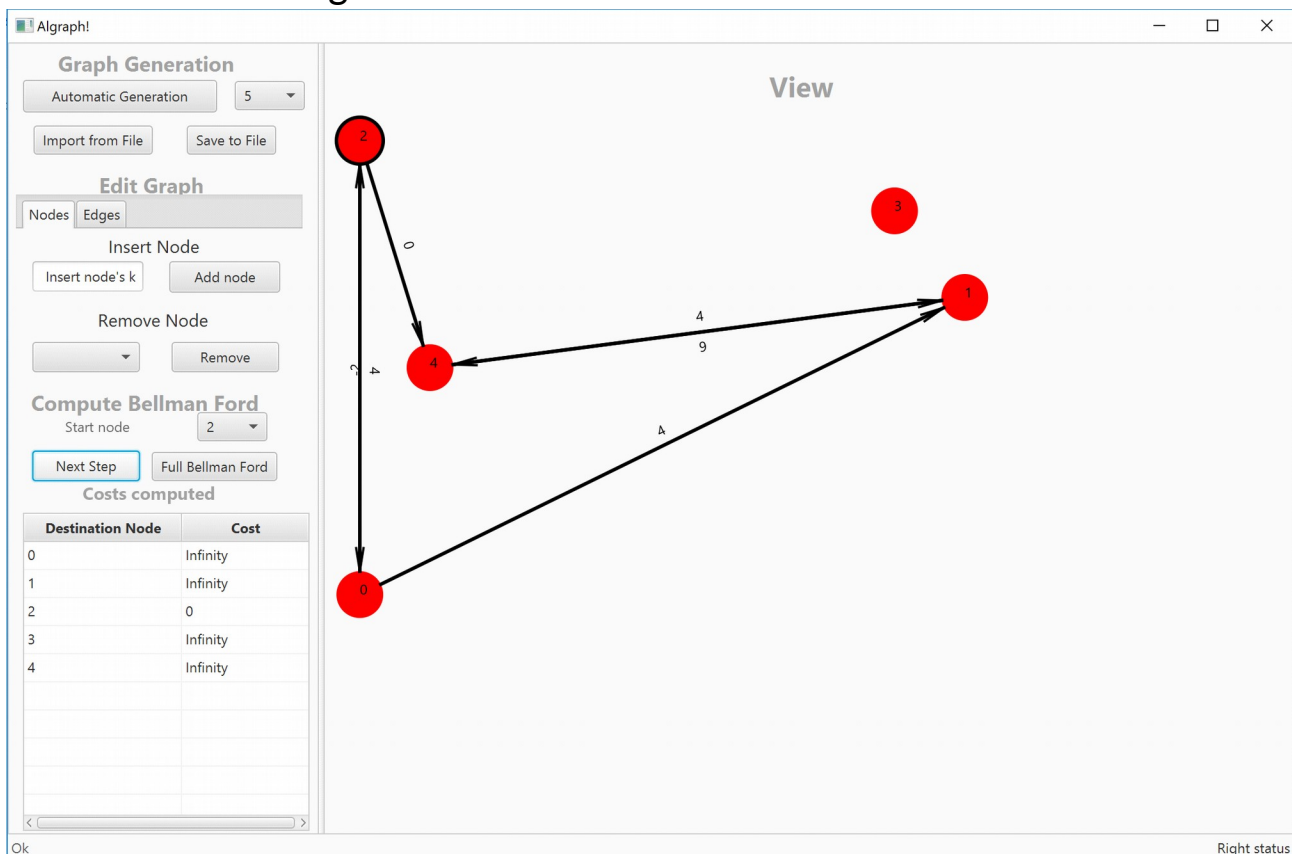


[l' arco che va dal nodo 0 al nodo 1 ha costo 1 mentre l' arco inverso (1, 0) ha costo 2]

Animazione algoritmo

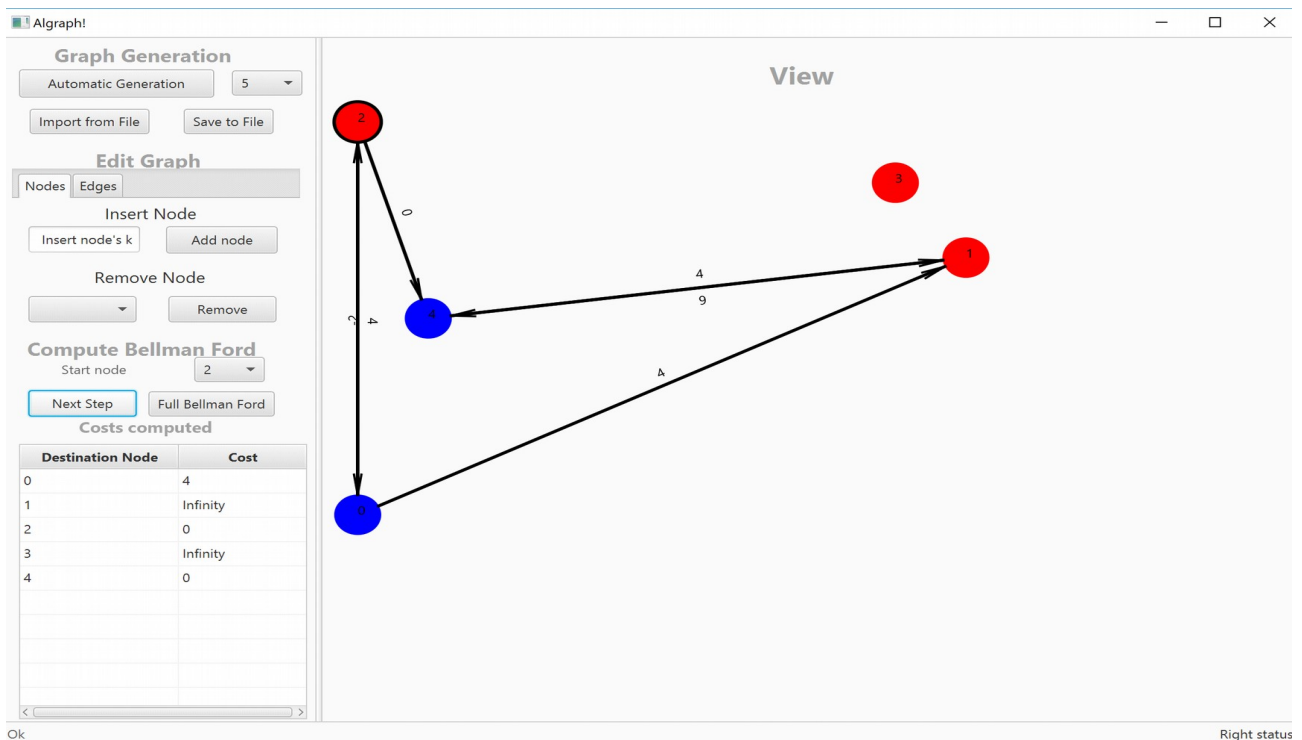
Per quanto riguarda l' esecuzione dell' algoritmo abbiamo creato una tabella che riporta per ogni nodo il relativo costo e come questo cambi ad ogni passo di Bellman-Ford (se ovviamente nel passo $d[v] > d[u] + w(u,v)$, altrimenti non succede nulla).

Inoltre il nodo da cui parte la computazione del peso degli archi è evidenziato nel grafo da un bordo nero e i nodi per i quali il costo cambia verranno colorati in Blu rendendo così evidente anche nel grafo l' animazione dell' algoritmo.

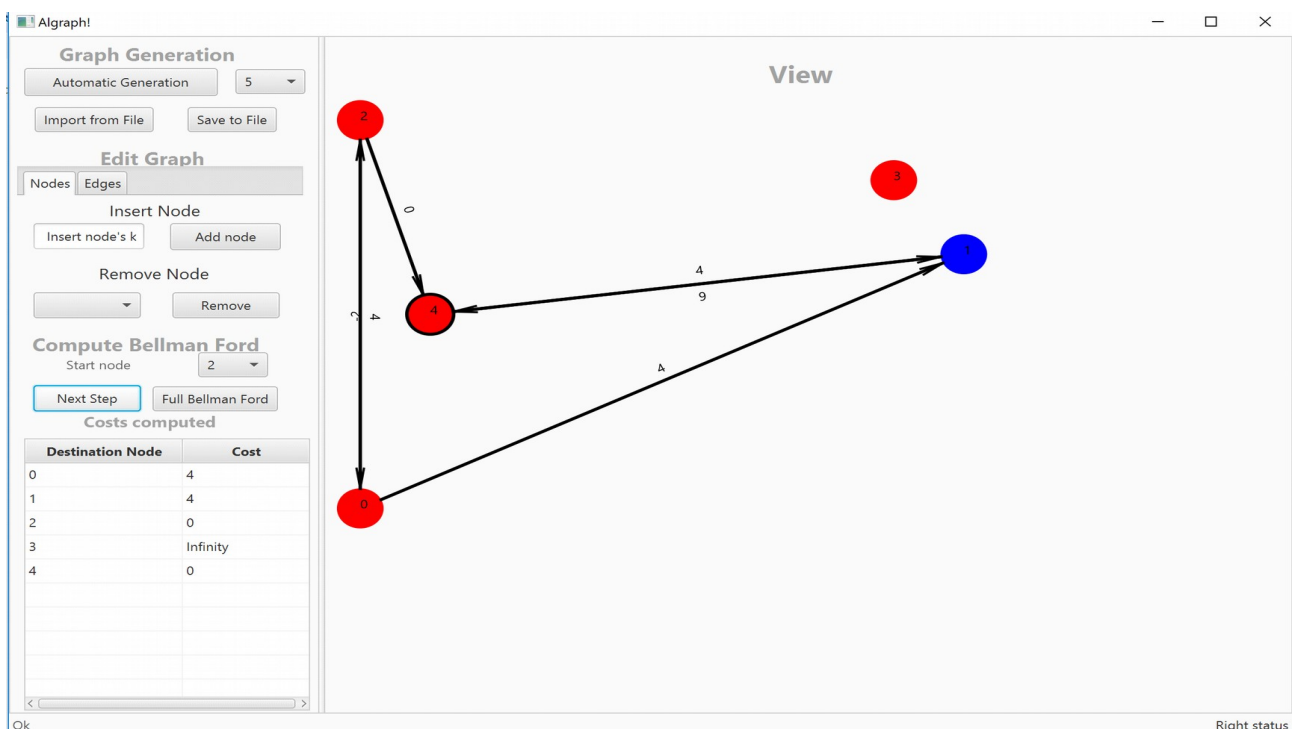


[il nodo di partenza è il 2 (evidenziato dal bordo nero), all' inizio della computazione il costo per raggiungere gli altri nodi è

"infinity"]



[dal nodo 2 vengono esaminati i nodi 4 e 0 che vengono colorati in blu e il costo aggiornato viene riportato nella tabella]



[il nodo da cui parte il rilassamento degli archi è ora il 4, quello in esame è l' 1. Il nodo 3, isolato, manterrà costo ∞]

Come da specifiche, infine, è possibile per l' utente modificare il grafo aggiungendo o rimuovendo un nodo/arco, aggiornare i pesi, generare il

grafo tramite input da file e salvarlo (se v'è almeno un nodo) in un file.txt
e infine computare l' algoritmo step by step o in un unico passo.

Daniele Ventura, Francesco Codicè