Relazione del 3° e 4° esercizio Laboratorio di Algoritmi e Strutture Dati

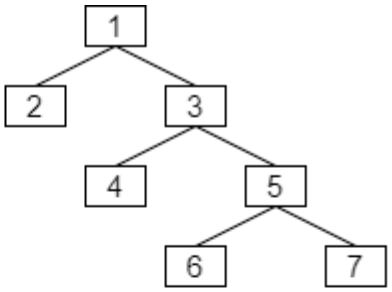
Zhao Xiao matr: 913828 mail: xiao.zhao@edu.unito.it

L'esercizio 4 si compila con il commando <ant>, e si usa il commando <java -jar build/MainKruskal.jar "./italian_dist_graph.csv"> per avviare.

La struttura di UnionFindSet è formato da 2 hashmap e usano entrambi il nodo come key, e rispettivamente il parent e rank come value.
Il findSet ritornerà il parent di un dato nodo e riduce l'altezza dell'albero in seguente modo:

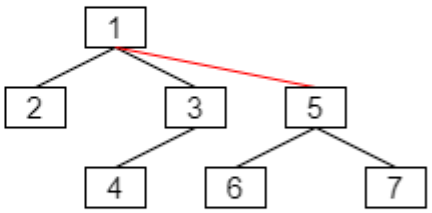
Struttura iniziale:

1	2	3	4	5	6	7
1	1	1	3	3	5	5



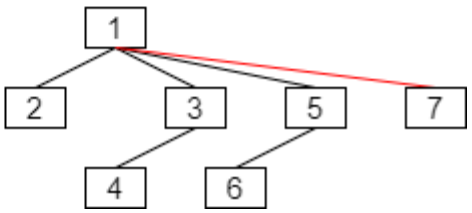
Si esegue il findSet su 7, ed esegue il findSet in modo ricorsivo fino ad 1 che ha il parent uguale a sé stesso. Poi ritorna ad 5 e modifica il suo parent ad 1.

1	2	3	4	5	6	7
1	1	1	3	1	5	5



Alla fine, modifica il parent del 7, riducendo l'altezza dell'albero.

1	2	3	4	5	6	7
1	1	1	3	1	5	1



Nella struttura Graph ho creato una lista di adiacenza usando 2 hashmap una dentro l'altra. Ho scelto di usare questa struttura perché hashmap permette di fare una ricerca o una visita a un particolare chiave in tempi minimi vicino a un costante O(1). Con questa caratteristica, alla chiave assegno il nodo che può essere string o integer, e riesco a soddisfare i requisiti richiesti dalla consegna.

Per la funzione getArchs() che recupera degli archi del grafo, ho creato una struttura Arch di supporto che ha come attributi i 2 nodi, il peso e una funzione compareTo per ordinamento. Questa funzione ritornerà un ArrayList di Arch, il quale lo userò come un lista temp per ordinamento.