

# Laboratorio di Algoritmi e Strutture dati\*

16 Dicembre 2013

## ◇ Esercizio Kruskal – (4pt)

Realizzare l'algoritmo di Kruskal completando la seguente classe:

```
class MinimalSpanningTreeKruskal <V, E extends DoubleSupplier> {  
    /*  
     * Coppia di vertici rappresentante un arco nel grafo.  
     * Se nell'implementazione delle strutture dati si è  
     * già definita una struttura analoga, è possibile usare  
     * la propria versione.  
     */  
    public class Edge {  
        V v1;  
        V v2;  
        E data;  
    }  
  
    /*  
     * Restituisce l'albero di copertura minimo del grafo g.  
     */  
    public List<Edge> build(UndirectedGraph<V,E> g) {  
        ...  
    }  
}
```

`DoubleSupplier` è una interfaccia definita dalla libreria standard in Java 8. Chi avesse deciso di usare Java 7, può definire `DoubleSupplier` come segue:

```
public interface DoubleSupplier {  
    double getAsDouble();  
}
```

`UndirectedGraph` è una versione non orientata della struttura dati `Graph`. La classe può essere implementata estendendo `SparseGraph` (o `DenseGraph`) e ridefinendo i metodi di accesso agli archi (e ai pesi) tenendo conto che le coppie  $(v_1, v_2)$  e  $(v_2, v_1)$  rappresentano lo stesso arco.

---

\*Revisione n. 1.0 del 26 maggio 2015

**Suggerimento:** È possibile ridefinire `addEdge(v1,v2,data)` in modo da aggiungere sia l'arco  $(v_1, v_2)$ , sia l'arco  $(v_2, v_1)$ . Se si procede in questo modo il resto dei metodi può essere lasciato inalterato. Sebbene questa soluzione non sia efficiente in termini di spazio (ogni arco e ogni peso sono rappresentati due volte), essa permette di minimizzare le modifiche alla classe base mantenendo inalterata la complessità dei metodi.