## Prim’s Algorithm

Prim's Algorithm is a greedy algorithm that is used to find the **minimum spanning tree** from a graph. Prim's algorithm finds the subset of edges that includes every vertex of the graph such that the sum of the weights of the edges can be minimized.

Prim's algorithm starts with the single node and explores all the adjacent nodes with all the connecting edges at every step. The edges with the minimal weights causing no cycles in the graph got selected.

In our case we needed to implement a minimum spanning forest, in other words we needed to find every minimum spanning tree from a graph. In order to achieve this, we decided to use the DFS visiting algorithm with its coloring method.

We also added a generic auxiliary class called Node<V, L> to maintain information about each vertex’s cost and whether they've been visited by the DFS or not, in order words it’s been used by both algorithms Prim and DFS visit.

To implement Prim’s minimum spanning tree algorithm we had to use minimum Priority Queue which was implemented by us for the third exercise. Priority Queue was implemented by min-heap data structure with combination of hashmap to keep certain method’s time complexity to constant.

At the beginning of Prim’s algorithm we need to initialize the costs of all the vertices except the first one to infinity. In order to set generic L type cost to infinity we used getInfinity method in Node class which returns maximum available number value depending on the L type (int, double, float etc). This can be achieved thanks to the fact that L extends Number class. Then we proceed to enqueueing the start vertice to the Priority Queue and start the loop and continue it until the Priority Queue is empty. At the beginning of each cycle we extract the first vertice from the Priority Queue that has minimum cost value and if there exists a neighbor we check its cost if it's higher than the edge cost between the chosen vertex and the neighbor vertex. If the cost of the edge is less than the vertex’s cost then we update it.

After successful execution of the minimum spanning tree (forest) we’ll have a subset of edges that includes every vertex of the graph such that the sum of the weights of the edges are minimum.

Overall the time complexity of the algorithm is O((|V| + |E|) log |V|) = O(|E| log |V|).