1. Remote searcher, local nodes: call of getDistance method on remote searcher requires nodes (the entire graph) to be transfered over network. Because NodeImpl is not remotely accessible it is serialized.
2. Local searcher, remote nodes: local searcher accesses the nodes by remote reference. This remote reference does not contain any data. So when getNeighbors is called neighbor nodes are transfered from the server.
3. Remote searcher, remote nodes: remote searcher assesses the nodes by remote reference. When getNeighbors is called, request is sent to the client and from client back to the server which returns a set of nodes. This way is traversed back to client and finally server.

We have used random graph with 500 vertices and variable number of edges. Same graph for all mentioned methods has been generated. Following figure shows dependency of number of edges in a random graph and time of getDistance() call for randomly chosen nodes (for each methods the same nodes). For each number of edges calculation has been performed 10 times and we took the average.

**Legend:**

LSLN - Local searcher, local nodes

LSRN - Local searcher, remote nodes

RSLN - Remote searcher, local nodes

RSRN - Remote searcher, remote nodes

**Observation**: Totally local search (LSLN) is the most efficient as nothing is transfered over network and nodes are accessible directly. Almost the same case is the RSLN method where the graph is transfered over network and the searching is performed locally (but on the server). This method would be useful if server is a high performance computer while client is just regular personal computer. The performance of the last two methods (LSRN and RSRN) is decreased by the fact that every step to the next node in the backtracking algorithm is a call over network.

Measured data:



We have also measured times of