To: Andrew P. Black
From: Bogdan Kovch
Date: May 16, 2013
Subject: A proposal for research of analysis of sorting algorithms

Project Topic: Sorting Algorithms
Sorting is a common way to organize such extensively used data structures like arrays and lists. Data sorting procedure is often required for other algorithms such as searching, inserting and deleting involved in data structure management. There are many sorting algorithms which differ in their behavior, input preferences, time and space efficiency, stability, and implementation complexity. Efficiency specifics are the most crucial while operating with data of large sizes. Three possible algorithms considered for this research will be selected from the following ones: Selection Sort, Bubble Sort, Insertion Sort, Mergesort or Quicksort.

Implementation Language: Java
Java is a widely used professional high-level programming language with the following advantages.

1. *Automated memory management* makes it very fast and easy to implement complex data structures and data manipulating algorithms such as sorting.
2. Availability of *developer-friendly* interactive IDEs with real-time debugging features which allow the programmer to focus on the subject of programming and avoid dealing with many distracting technical inconveniences.

Features in Final Report.
The final report will include the analysis of several sorting algorithms implemented in the Java programming language. The following features will be used to perform comprehensive analysis.

1. ***Analysis Tools.*** Each algorithm will have a wrapping or built-in code for collecting statistics necessary for analyzing algorithm's behavior. Some of them are as follows:
   a) count of times a particular basic operation was performed;
   b) overall time taken to produce an output for a specific input;
   c) average time needed to process one element in the input sequence;
2. ***Testing Strategy.*** The programs will generate random or load from file specific sequences of values of primitive data types to be fed into an algorithm as input.
   a) *Specific data sequences* needed to test the best-case and worst case efficiency. The most common examples are sequences of numbers already sorted in ascending or descending order or other sequences which will allow to demonstrate specific properties of an algorithm.
   b) *Random data sequences* will be generated by the program in order to test an average-case efficiency of an algorithm. In order to do this, a number of different random sequences will be fed into the sorting algorithms. Their statistics will be combined to produce an average efficiency statistics.
3. ***Analysis process and results.*** The report will contain comments describing the analysis process, implementation details of algorithms and analysis tools, the specifics of the test cases with corresponding behavior statistics, summary and experience gain during this research. Many materials will be presented via tables, charts, or other visual means.

Time Plan.

1. ***Week of 23rd May****:* Implement all sorting algorithms and create specific test cases for them.
2. ***Week of 30th May:*** Implement algorithm analysis tools and random sequence generator.
3. ***Week of 6th June:*** Write the final report based on algorithm tests and analysis output.

Collaboration Plan.
This will be an individual research project.