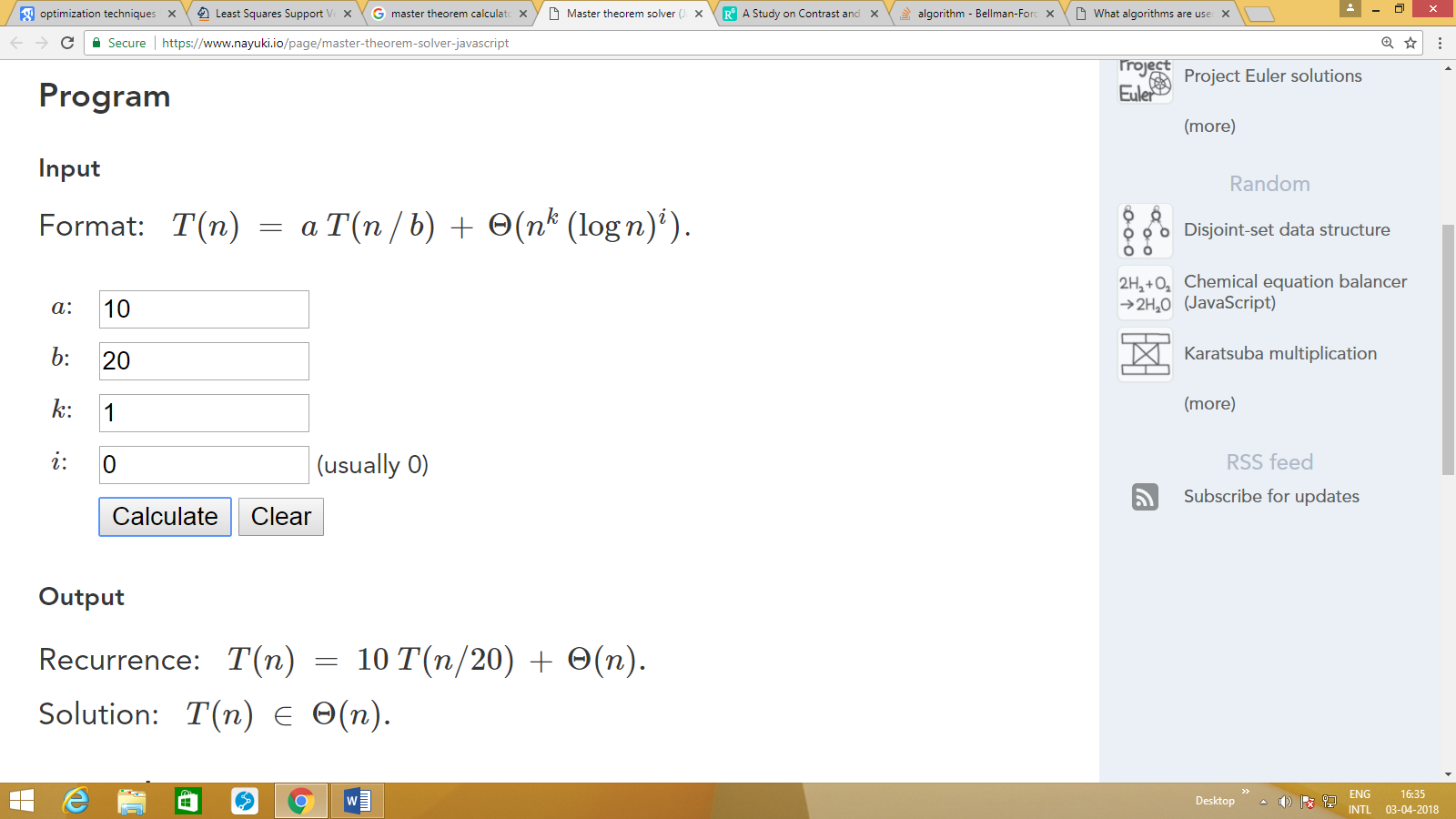
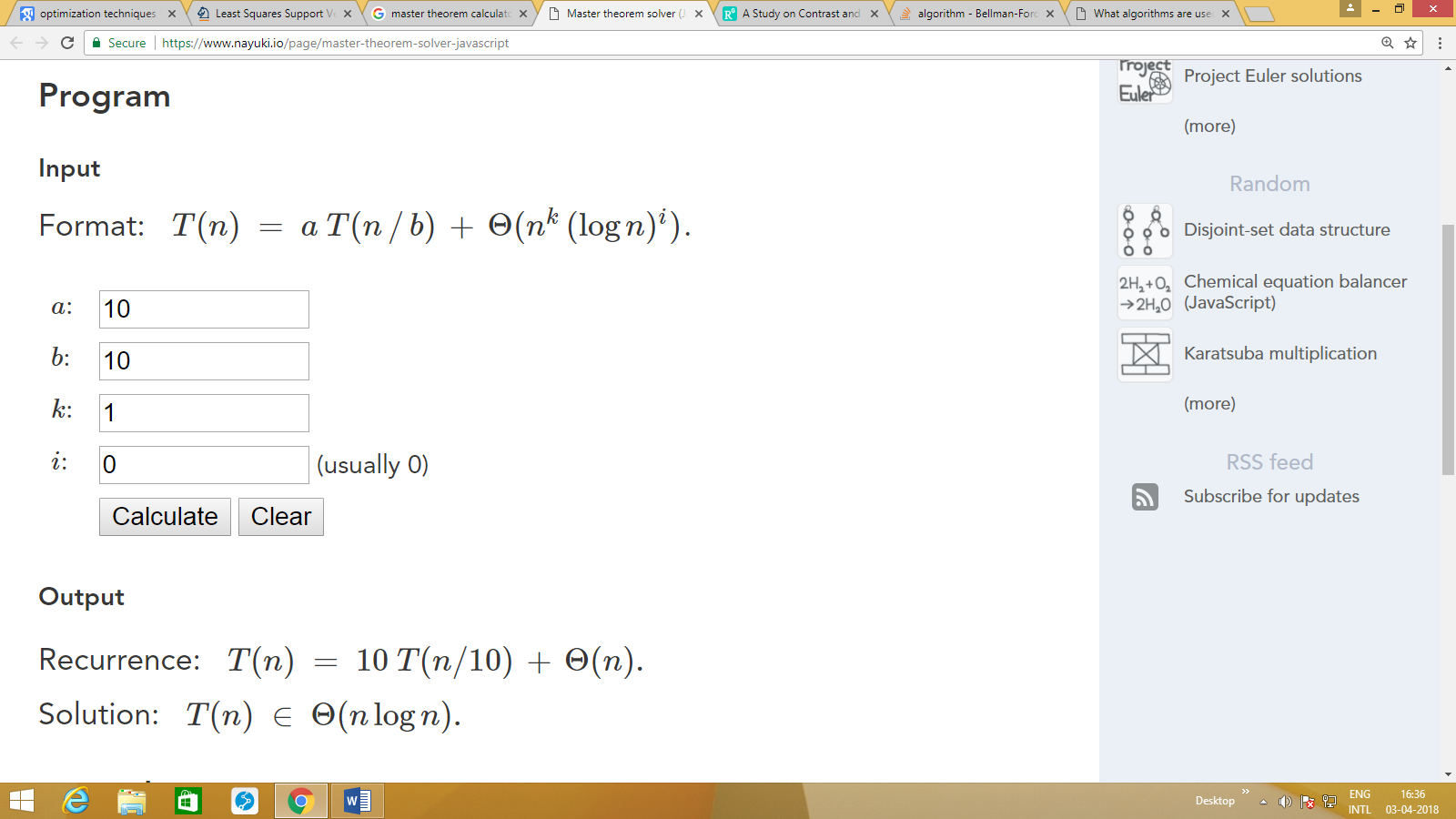
**MASTER METHOD**

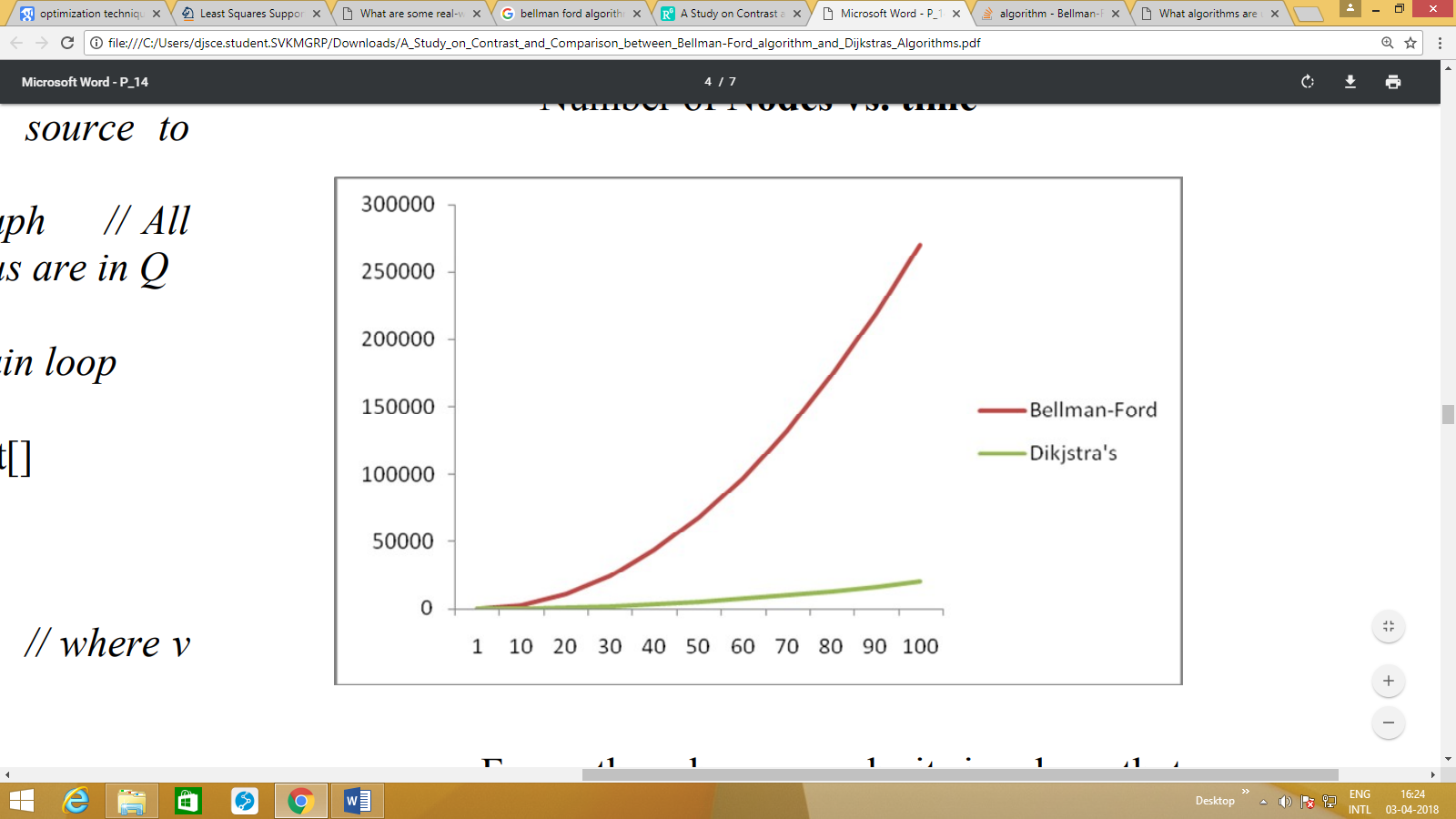


OPTIMIZATION FOR SHORTEST PATH BETWEEN A HIGH NUMBER OF NON NEGATIVE NODES USING DIJKSTRA’S ALGORITHM IN PLACE OF BELLMAN FORD  
Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, outputting a shortest path tree. This algorithm is often used in routing. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined.

The Bellman–Ford algorithm, sometimes referred to as the Label Correcting Algorithm, computes single-source shortest paths in a weighted digraph (where some of the edge weights may be negative).Bellman-Ford is in its basic structure very similar to Dijkstra's algorithm, but instead of greedily selecting the minimum-weight node not yet processed to relax, it simply relaxes all the edges, and does this |V| - 1 times, where |V| is the number of vertices in the graph. The repetitions allow minimum distances to accurately propagate throughout the graph, since, in the absence of negative cycles, the shortest path can only visit each node at most once. Unlike the greedy approach, which depends on certain structural assumptions derived from positive weights, this straightforward approach extends to the general case.

Form the above graph it is clear that though the nature of the two curves are same i.e O(n2 ), the Bellman ford algorithm requires more time than Dijkstra’s algorithm.

RIPv2 uses Bellman Ford whereas OSPFv2 uses Dijkstra’s algorithm.



As the analysis shows the Bellman-Ford algorithm soles a problem with a complexity of 27n2 but the Dijkstra's algorithm solves the same problem with a lower running time, but requires edge weights to be non-negative. Thus, Bellman– Ford is usually used only when there are negative edge weights. Both of these functions solve the single source shortest path problem. The primary difference in the function of the two algorithms is that Dijkstra's algorithm cannot handle negative edge weights. Bellman-Ford's algorithm can handle some edges with negative weight. It must be remembered, however, that if there is a negative cycle there is no shortest path.

**CODE COMPLEXITY OF MERGE SORT**

