Biconnected Components Algorithm Performance Analysis

`

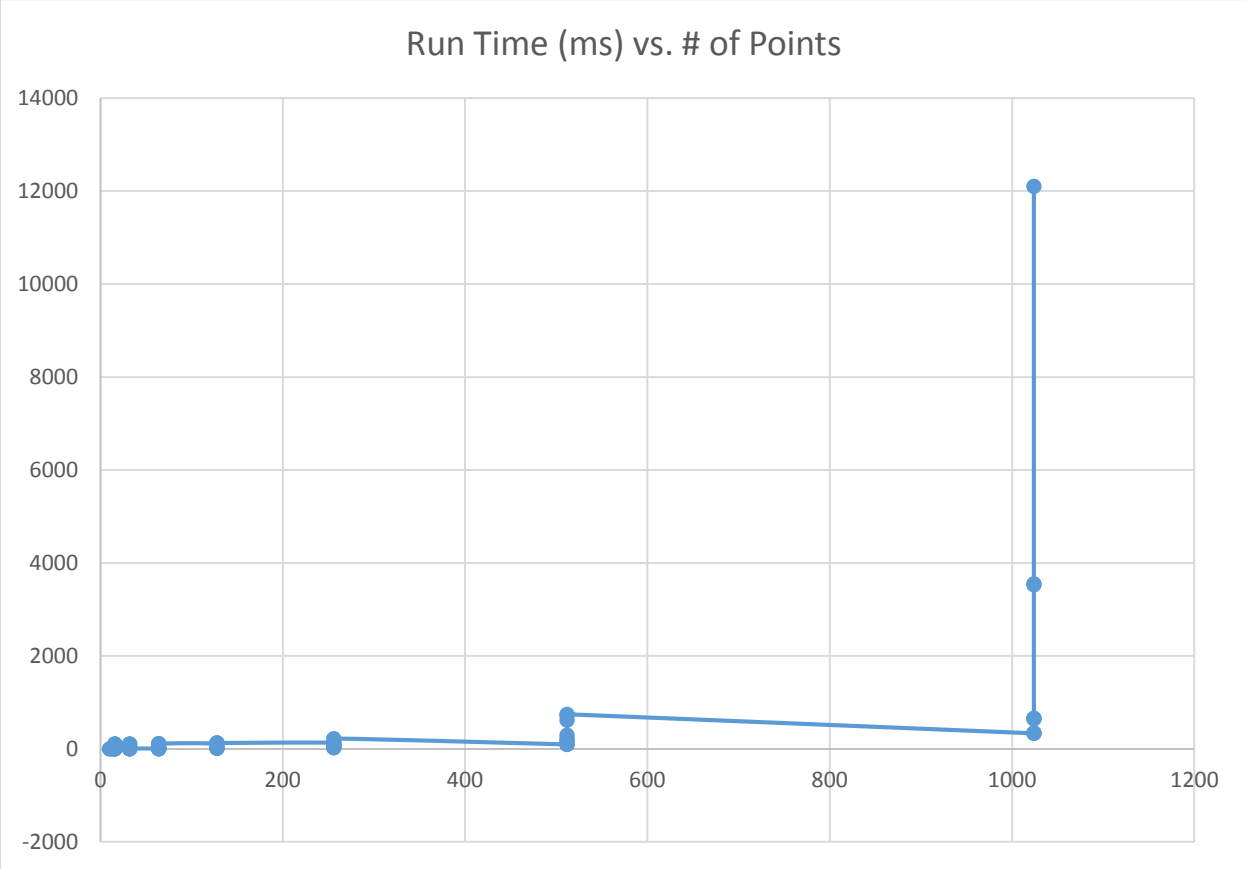
ZHANG, YANG 1030146

CSE 417

From my measurements, I finalized that the runtime of Biconnected components algorithm is raised when the size of problem increasing in log scale.

However, the runtime of this algorithm is not only depends on the number of the points but also the number of the edges.

The following figure shows Runtime vs. the number of points:



From the above the graph, the runtime is basically increased when the size increased, but there are few points shows that the runtime of 1024 points is close or even faster than mush smaller size.

The following figure shows the Runtime vs. Number of edges:

From this plot, we can clearly say that the run time of Biconnected component algorithm is proportional to the number of edges. The difference of runtime is not significant when edge numbers are relatively small, however, when the edge number getting bigger the runtime is increased greatly.

As a result, the number of edges is the direct factor to the runtime of Biconnected components algorithm.

The theoretical big-O of this algorithm is O(v+e), which v is the number of vertexes and e is the number of edges.

From my measurement:

Vertexes	Edges	Runtime(ms)
1024	3090	296.7184
1024	31510	3549.731

Vertexes	Edges	Runtime(ms)
256	754	110.8949
128	752	27.014

I can conclude that for the same number of vertexes, the more edges the graph has, and the more runtime it need. While for the same number of edges, the more vertex the graph has, and the more runtime it need. This result is consistent with theoretical big-O analysis.

System Environment:

1.8 GHz Intel dual core i5 with 256KB L2 cache and 3MB L3 cache

4 GB 1.6GHz DDR3 Memory

Test Case Result:

file name: testcase.txt

*** Number of nodes: 16

*** Number of edges: 30

*** Number of Biconnected Components: 5

*** Number of Articulation Points: 4

*** List of Articulation Points: ['1', '12', '5', '8']

*** List of Biconnected Components:

BC1: [['1', '10'], ['10', '14'], ['14', '15'], ['2', '15'], ['1', '15'], ['3', '15'], ['1', '3'], ['2', '3'], ['1', '2']]

BC2: [['1', '12']]

BC3: [['5', '12']]

BC4: [['5', '8']]

BC5: [['6', '11'], ['7', '11'], ['11', '13'], ['9', '11'], ['0', '9'], ['9', '13'], ['7', '13'], ['6', '13'], ['8', '13'], ['6', '8'], ['4', '8'], ['0', '8'], ['7', '8'], ['0', '7'], ['6', '7'], ['4', '6'], ['0', '4'], ['0', '11']]