

Analysis of an Eulerian Circuit Finding Algorithm

Anthony Naddeo

Index Terms—Eulerian, algorithm, graph, circuit

I. INTRODUCTION

Our assignment this week asked us to create an algorithm that finds Eulerian circuits in graphs. The purpose of this paper is to prove that the running time of our algorithm is indeed $O(n)$ by providing empirical evidence in the form of run times from our implementations.

II. RESULTS

In the written section of this assignment I show that the time complexity of this Eulerian circuit finding algorithm is $O(n)$. To provide an empirical proof, I wrote a test program that generates simple Eulerian circuits of any given size n , where n represents the amount of edges in the graph. The output would be fed to the Java application and the running time recorded with the `System.currentTimeMillis()` function, taking the average run time from five executions with each n tested. The test was split into two timed sections: the amount of time that Jflex and Cup took to parse the input and the amount of time the application took to find the Eulerian circuit from the generated graph. Figure 1 shows the results of the circuit finding algorithm.

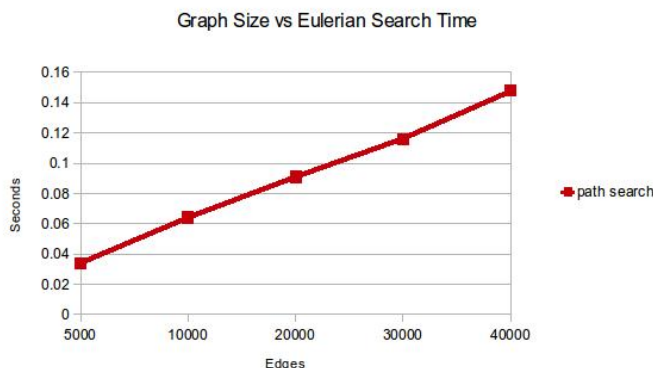


Fig. 1

As previously shown, the time complexity of the algorithm is $O(n)$; it grows linearly as n increases. The core of the algorithm consists of a single loop that iterates once for every edge in the graph. The success of this algorithm is contingent

```
Vertex current, next;
while (current.degree() > 0) {
    next = current.getEdges().get(0);
    Vertex.disconnect(current, next);
    solution.addStep(current.name(), next.name());
    current = next;
}
```

Fig. 2: Main loop of eulerian circuit algorithm

upon the data structures holding the graph vertices being sorted in descending order.

Perhaps of greater interest is the performance of the JFlex and Cup parser. Figure 3 shows the time that parsing took as n increased. The program first parses input and then finds the resulting circuit; each parse time was recorded right before a corresponding circuit finding time.

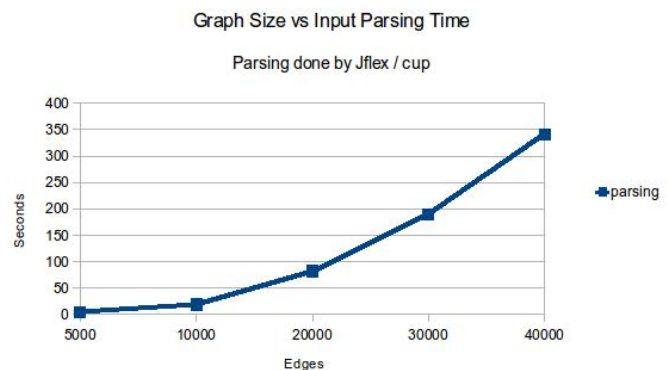


Fig. 3

The parser appears to have a polynomial time complexity. Whether this is expected behavior or inefficiency on my part is unknown.

III. CONCLUSION

The results posted in this paper support the previous statement that the algorithm created for this assignment has a time complexity of $O(n)$.