Visualize Algorithms: Complex analysis of the Fibonacci heap

Qingjie Lu

Introduction

Fibonacci heap is a collection of trees in computer science. It has better amortized analysis performance than the binomial heap and can be used to implement merge priority queues. Operations that do not involve deleting elements have an amortized time of O(1). The number of Extract-Min and Delete is more efficient than the others.

A Brief Survey of Known Results

For most students, the learning process of algorithm is boring, but we can make the algorithm move more vividly through visualization. By visualization, students can not only exercise their programming ability, but also vividly understand the advantages and disadvantages of each algorithm.

Although there are many applications and software to introduce the dynamic process of algorithm, we still stay in the stage of watching and learning and do not really implement the specific algorithm and application.

A Potential Plan and Idears

So I prepared four or five steps to implement the proposal. First, I chose Java GUI for graphic programming. Using Java.swing module to achieve graphics rendering, graphics movement, mouse events, keyboard events and other functions. This step can be replaced by other languages. In the second step, I want to test the feasibility of visualization by some simple sorting algorithms, and simply compare those sorting algorithms. It includes selection sort, insertion sort, merge sort, quicksort and heap sort. Third, I implemented two simple

applications: maze and automatic maze generation. I chose DFS, BFS, and non-recursive DFS to implement this two application. The first three steps are mainly to pave the way and explore the feasibility of the project; GUI preparation, algorithm performance comparison, algorithm application. In step 4, I will visualize the time complexity analysis of the mergeable operations of the Fibonacci heap versus the binomial heap. Step 5, I'll try to use the Fibonacci head to implement a small application such as single-source shortest paths, weighted bipartite graph matching, and first-order tree problems.

References

Michael L. Fredman and Robert E. Tarjan. Fibonacci heaps and their uses in imporved network optimization algorithms. *Journal of the ACM*, 34(3):596-615,1987

https://bost.ocks.org/mike/algorithms/

https://www.cs.usfca.edu/~galles/visualization/Algorithms.html