

Title of research:

Development of heap structure for data storage and output in Clojure

Name: Zhu Jiarui

Date: 2021/11/30

Background and Justification:

Heap is a kind of efficient data storage structure, especially in the representation of priority queues. So it is a good choice for data logging, sorting and output.

Existing implementations are mostly based on index operations in vectors, rather than the real tree structure. This kind of structure cannot meet the requirements for efficient computing.

In this case, we need a maximum/minimum heap implementation scheme that is more in line with the characteristics of the heap.

Aims and Objectives:

Develop a heap structure to force the order of the result file.

Implement basic heap APIs (init, add, and pop) in max or min-heap.

Methodology:

Do research on existing libraries to find out the implementation logic of heap.

Learn Clojure and look for data structures and operations that can help.

Design and implement heap structure and functions.

Use the designed test data and real data to test the performance and effectiveness of the method.

Plan of Action:

Week 1: Do Clojure study. Find out useful information.

Week 2: Finish basic binary tree node implementation.

Week 3&4: Roughly realize the functions needed by the heap.

Week 5: Test and Debug. Improve the function.

Week 6: Real data testing and interface implementation.

Week 7: Interface debugging.