

## **Research Project Report**

Mar 2022

### **Research project:**

Development of heap structure for data storage, sorting, and output in Clojure

### **Name:**

Zhu Jiarui

### **Research Methodology:**

- Use Clojure as the main development language.
- Provide an interface for using PriorityQueue in Java to
- Imply the sort, push, pop operation of the heap through pure Clojure sentences.
- Compare the performance of Java PriorityQueue-based heap and Clojure record-based heap.

### **Research Background:**

Heap is an efficient and classical data sorting and storage structure. However, there is no package in the current Clojure open-source code to support the heap structure. Therefore, we need to find some ways to imply heap in order to improve the input and output efficiency of large dataset handling in Clojure.

### **Research Finding:**

The first part of the research is to import PriorityQueue in the Java library and help Clojure users to do heap operations.

For the second one which is developing heap using Clojure, the current version of the Clojure heap can complete continuous pushing and popping operations within  $O(\log(n))$  running time. Each heap supports increasing and decreasing sorting order.

Since Clojure does not have an efficient method for partially modifying objects, the current version can only complete heap updates by constantly constructing new objects and replacing old ones, which might be a huge cost of storage space.

**Research Limitation:**

Due to the lack of in-depth understanding of Clojure, the current version does not implement complete encapsulation, but only a few functions that can complete operations. Also, the current version does not provide heap balancing, which can be completed in future studies.