

CS 36C Winter 2024 Homework 3: SkipList and Sorting

This project has you writing an implementation of a SkipList (to do a key/value data store), mergesort on linked lists, and quicksort on arrays. This assignment is due on Tuesday February 6th at 1400 PST (6:00 pm Davis time). You must work on this assignment individually. We have provided you with a rich skeleton project in **Homework3.zip** available from Canvas which includes a full Gradle and IntelliJ environment.

This time the Skeleton project actually includes **all** the tests used for grading, as a sufficient number of students submitted photos for Cat Break.

Expected Project Outcomes

1. Implement the major functionality of a SkipList, including the iterator (which produces pairs), setting, and getting elements.
2. Implement mergesort for a LinkedList
3. Implement quicksort for arrays

Implementation and Grading

You should implement the specified function in the three Kotlin files. These are:

1. `SkipListCell.makeNextArray(): Array<SkipListCell<K, V>?>`
2. `SkipListIterator.computeNext(): Unit`
3. `SkipList.set(): Unit`
4. `SkipList.getNode(): SkipListCell<K, V>`
5. `SkipList.contains(element: <K>): Boolean`
6. `LinkedList.split(start: LinkedListCell<T>) : Pair<LinkedListCell<T>, LinkedListCell<T>>`
7. `LinkedList.mergeSortWith(comp: Comparator<in T>) : LinkedList<T>`
8. `Array<T>.quickSortWith(comp: Comparator<in T>) : Array<T>`

For grading on Gradescope you will submit `SkipList.kt`, `LinkedList.kt`, and `ArraySorter.kt` files.

We have provided you with the comprehensive test suite used for grading, but you may add further tests if you desire and are encouraged to do so.

The SkipList basic operations (insertion, fetching, and checking) should be $O(\log(n))$ and the two sorts should be $O(n \log(n))$ in all cases (in the case of QuickSort this means no pathological cases that are $O(n^2)$). The test cases include the performance evaluation necessary.