

Lab 1

1. Write code for insertion sort.
2. Write code for merge sort.
3. The running time of merge sort can be improved in practice by taking advantage of the fast running time of insertion sort when its input is “nearly” sorted. When merge sort is called on a subarray with fewer than k elements, use insertion sort to sort the subarray. Argue that this sorting algorithm runs in $O(f(n, k))$ expected time. What is $f(n, k)$ and how should k be picked, both in theory and in practice by experiments?
4. Write code for improved version of sorting algorithm which combines merge sort with insertion sort.
5. All document for the answers of the above questions.