Project 1

1. Write code for insertion sort. (10 points)

2. Write code for mergesort. (10 points)

3. Write code for quicksort. (10 points)

4. The running time of quicksort can be improved in practice by taking advantage of the fast running time of insertion sort when its input is "nearly" sorted. When quicksort is called on a subarray with fewer than k elements, let it simply return without sorting the subarray. After the top-level call to quicksort returns, run insertion sort on the entire array to finish the sorting process. Argue that this sorting algorithm runs in $O(nk + n\ lg(n/k))$ expected time. How should k be picked, both in theory and in practice by experiments? (30 points)

5. Write code for improved version of sorting algorithm which combines quicksort with insertion sort. (20 points)

6. All document for the answers of the above questions. (20 points)