• (0 points (*) Problem 3-3 on page 61 of the textbook.
• (0 points (*) Problem 3-4 on page 62 of the textbook.
• (0 points) Problem 4-1 on page 107 of the textbook.
• (100 points)
  1. Implement merge sort.
  2. Implement insertion sort.
  3. Implement a combination of insertion and merge sort as described in Problem 2-1 on page 39 of the textbook.
  4. Develop a suitable test strategy (e.g., size of numbers to be sorted, amount of numbers to be sorted, number of test runs for each instance) that will allow you to properly test and compare the performance of your implementation of these three algorithms. You must provide a detailed write-up of your test strategy, including a discussion why you consider this test strategy to be appropriate.
  5. Conduct the tests. Provide a detailed write-up and analysis of your test results (including plots of timings etc.).
  6. Aside from the write-ups your submission must include the code, makefiles, your code for the testing, the scripts that allow to re-generate your tests, all test data that was used for the analysis, scripts to generate your graphs etc.
  7. The assignment must be completed in C/C++ (unless otherwise stated). Your code must compile on the linux-lab machines. Programs that do not compile will receive an automatic grade of zero (0).

(*) means that you are expected to do the problems but you will not submit solutions and you will not be graded on these assignments.