

## Objectives

In this lab, you will implement the Quick Sort algorithm and compare its running time to the basic sorting algorithms implemented earlier in Lab 5.

## Deliverables

- (10%) You have to attend the lab on Monday.
- (50%) The final source code.
- (5%) You must adhere to the following submission format. You need to submit a single file on iLearn named “CS014\_lab4\_<UCR Net IDs>.zip” where <UCR Net IDs> are the students’ UCR Net IDs separated by underscore. The ZIP file should contain the source code in .cc, .hh, or whatever extension for source files you. The ZIP file should also contain a PDF report that contains the filled-in table (see below) and the log-log chart for the running times of the algorithms. On the cover page, your report should mention:
  - Your TA name.
  - Your lab section number.
  - You name and UCR Net ID
- (15%) Report the final running times in a table and plot a log-log chart.
- (20%) The in-lab assignment on Monday 11/13.

## Groups

- This lab will be done individually.

## Due date

- The deliverables are due on Tuesday 11/14 by 11:59 PM Pacific Time. However, you are highly encouraged to deliver it during the lab on Monday 11/13 to save your time.

## Problem definition

In the previous lab, you implemented four basic sorting algorithms. In this lab, you will implement the Quick Sort algorithm and compare its running time to the previous ones.

## Steps

1. Use the same workspace that you used Lab 5. You can do this by pasting the URL ‘<https://github.com/aseldawy/CS014-Lab5.git>’ in the field ‘Clone from Git or Mercurial URL’

when you create a Cloud9 workspace. Instead, you can download the source code from iLearn and upload it to an empty workspace in Cloud9.

2. Add a function 'quickSort' that has the same signature as the other sorting algorithms.
3. Expand the table that you used in Lab 5 by adding a column for Quick Sort. Run the Quick Sort algorithm and compare its running time to the other algorithms.

| Array size | Bubble Sort | Insertion Sort | Selection Sort | Shell Sort 1 | Quick Sort |
|------------|-------------|----------------|----------------|--------------|------------|
| 1          |             |                |                |              |            |
| 2          |             |                |                |              |            |
| 4          |             |                |                |              |            |
| 8          |             |                |                |              |            |
| 16         |             |                |                |              |            |
| 32         |             |                |                |              |            |
| 64         |             |                |                |              |            |
| 128        |             |                |                |              |            |
| 256        |             |                |                |              |            |
| 512        |             |                |                |              |            |
| 1024       |             |                |                |              |            |
| 2048       |             |                |                |              |            |
| 4096       |             |                |                |              |            |
| 8192       |             |                |                |              |            |
| 16384      |             |                |                |              |            |
| 32768      |             |                |                |              |            |
| 65536      |             |                |                |              |            |

4. Update the log-log plot by adding Quick Sort to it.
5. Comment on the behavior of the Quick Sort algorithm as compared to the basic sorting algorithms.