COSC1285/2123: Algorithms & Analysis
Laboratory 3

Topic

Sorting.

Objective

Students who complete this lab should:

- Learn to implement a fundamental sorting algorithm.

Introduction

In this lab exercise you are to implement selection sort and compare its performance to bubble sort.

Provided Code

`SortDemo1.java` reads a sequence of white-space separated integers from file and sorts them into ascending numerical order. `SortDemo1.java` allows you to sort the input using several different sorting algorithms. The algorithms were covered in the lecture and in the text book.

The program is divided into the following modules:

file	description
`SortDemo1.java`	Code to read data from disk into the set. Also performs timings. No need to modify this file.
`BubbleSort.java`	Implementation of standard bubble sort. No need to modify this file.
`SelectionSort.java`	Your implementation of selection sort.

Compile the program using the following command:

```
javac *.java
```

Run the command using the following parameters:

```
$ java SortDemo1
USAGE: SortDemo1 [sort method] [input file]
  sort methods [bubble, selection]
EXAMPLE: SortDemo1 selection random.txt
```

The `SortDemo1.java` program supports the following algorithms:

algorithm	description
`bubble sort`	Standard bubble sort algorithm.
`selection sort`	Standard selection sort algorithm.

Task

Your task in this lab exercise is to implement the following sorting algorithm:

Selection sort is to be implemented in SelectionSort.java.

After implementing selection sort, run each of the algorithms (bubble, selection) on the following test files and compare the run times:

file	description
`debug.txt`	file with few items for testing purposes.
`random.txt`	file with items in random initial order.
`nearlysorted.txt`	file with items nearly ordered.
`reversed.txt`	file with items in reversed sorted order.
`fewunique.txt`	file with very few unique items