Data Structure

Lab 6. Sorting

Shin Hong

May 22, 2020

Lab 6 - Submission & Evaluation

- Submission deadline: 9 PM, May 28 (Thur)
- Online test: http://34.64.144.206 Session ID: QPgAjFY
- Evaluation: 5 points x 2 problem (total 10 points)
  - 5 points per problem: succeed on time (before the deadline)
  - $\leq$ 3 points: per problem: submit a report by 4 PM, May 29 (Thur)

Problem 1. Linked List Quick Sort

- Complete `linkedlist_qsort()` in the doubly linked list given at https://github.com/hongshin/DataStructure/blob/sorting/linkedlist
  - You must fill out the missing part marked with `TODO`, and you must not change the other parts
  - For your reference, selection sort is given at `linkedlist_sort()`

- The main function receives a positive integer *i* from the standard input and then prints the *i*-th word after the sorting
  - Auto-grading will check whether the main function produces the correct outputs for given test cases

Problem 2. Intervals

- Write a program that receives a set of real-value intervals and then finds the greatest number of intervals that are overlapped at a point
  - an interval $[b, e)$ is the set of all real numbers r such that $b \leq r < e$
  - a set of intervals are overlapped when there exists a real number is included in all intervals

- Use the given arraylist and its sorting function for constructing the solution program
  - https://github.com/hongshin/DataStructure/tree/sorting/arraylist

- Construct your solution as a single source file

Input and Output

- Input
  - Given from the standard input
  - First line gives the number of intervals N for $1 \leq N \leq 100$
  - Each of 2nd to (N+1)-th lines has two real numbers b and e that represents [b, e) for $-1000.0 \leq b < e \leq 1000.0$
- Output
  - Print an integer to the standard output, that is the greatest number of the intervals that are overlapped at a point
  - Make sure to put newline ('\n') at the end

Examples

```
5↵
15.5_20.3↵
-110.23_225.349↵
-10.9_0.75↵
19.5_30.11↵
150.0_325.6↵
```

<Input 1>

```
3↵
```

<Output 1>

```
4↵
-444.498_-141.8279↵
-306.062_97.7435↵
153.166_297.268↵
32.1586_100.6203↵
```

<Input 2>

```
2↵
```

<Output 2>

```
5↵
502.6_620.3↵
271.07_404.138↵
737.28_829.9↵
-371.2_502.6↵
-582.1_-371.2↵
```

<Input 3>

```
2↵
```

<Output 3>