실습과제 #3

1. 목표: list 자료구조에 sorting algorithm들을 구현하는 방법을 학습하고 이들의 실행시간들을 측정해서 비교 한다.
2. 문제 배경: 숙제 1에서 array를 이용하여 직원( Employee)들을 관리하는 프로그램을 만들었다.
3. 문제
   1. 이번에는 이들을 ADT list를 사용하여 직원들을 관리하는 프로그램을 만들고자 한다. 1번 프로그램숙제에서 사용했던 class를 재사용하라. 다만 socialSecurityNumber는 순차작으로 하고 나머지 자료는 의미가 없는 숫자 이므로 난수를 사용해서 발생 시킨다. 이들을 리스트에 넣으면 자동적으로 정렬이 된다.
   2. 직원의 자료가 정렬되어 있기 때문에 이것을 정렬이 안된 자료로 만들어야 한다. 이를 위해서 전반부 절반의 숫자를 난수로 만든 위치로 이동시킨다. 즉 swap한다.
   3. 교과서 12장에서본 ADT list에는 sorting하는 기능이 없었다. 이 기능을 추가 하고자 한다. 두 가지 sorting algorithm(insertion sort 과 quick sort)을 구현하라. 결과적으로 ADT List에 qSort()와 iSort()를 추가 하는 것이다. 교과서 13, 14장에서 array을 사용하는 방법과 linked-list을 사용하는 방법을 모두 구현 하였다. 여기서는 array인 경우만 한다.
   4. 이들 두 알고리듬의 실행 시간을 비교 하라. 실행 시간은 input size 별로 측정해서 비교 table이나 그래프를 만든다. Size가 100, 1000, 10000, 100000인 경우에만 측정하라. 시간을 측정하는 방법은 아래와 같다. 다만 중간에 io 즉 print()를 넣지 말아야 한다.

long startTime = System.currentTimeMillis();

여기에 sorting algorithm 을 넣으면 된다.

long endTime = System.currentTimeMillis();

elaspedTime = endTime - startTime;

1. 제출물:
   1. source program
   2. 실행시간을 측정해서 비교하는 도표나 그래프
   3. 실행한 결과물 screen Dump한 것