실습과제 #3

1. 목표: 2가지 sorting algorithm들을 구현하고, 이들의 실행시간들을 각기 다른 data size의 경우에 측정해서 비교함으로서 time complexity가 주는 의미를 확인해보고자 한다.
2. 문제 배경: 숙제 1에서 array를 이용하여 직원(Employee)들을 관리하는 프로그램을 만들었다. 숙제 2에서는 Bag를 사용해서 직원을 관리했다. 이번에는 다시 array를 사용해서 sorting algorithm들을 구현해보고자 한다.
3. 문제
   1. \*1번 프로그램숙제에서 사용했던 Employee class를 재사용하라. 다만 int ssn(socialSecurityNumber)변수부분을 추가하고, 그 값은 순차적으로 하고, 나머지 필드는 의미가 없는 숫자 이므로 dummy단어들을 사용해서 employee object들을 만든다. 이들을 array에 넣으면 자동적으로 정렬이 된다.
   2. \*또한, Employee class는 Comparable Interface의 compareTo() method를 구현해야 한다. 그래야 순서를 정할 수 있다. compareTo()에서 ssn 값이 큰 것이 큰 object가 되도록 해야 한다.
   3. \*직원의 자료가 정렬되어 있기 때문에 이것을 정렬이 안된 자료로 만들어야 한다. 이를 위해서 전반부 절반의 숫자를 난수로 만든 위치로 이동시킨다. 즉 swap한다.
   4. \*두 가지 sorting algorithm(insertion 혹은 selection sort 과 merge 혹은quick sort)을 구현하라.
   5. \*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