**INFO 6205**

**Program Structures & Algorithms**

**Fall 2020**

**Assignment No 2**

* **Task**

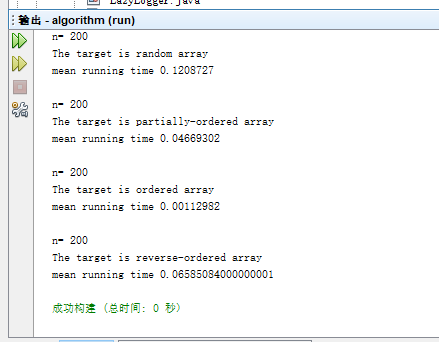
(Part 1) You are to implement four methods of a class called Timer. Please see the skeleton class that I created in the repository. Timer is invoked from a class called Benchmark\_Timer which implements the Benchmark interface.

(Part 2) Implement InsertionSort (in the InsertionSort class) by simply looking up the insertion code used by Arrays.sort. You should use the helper.swap method although you could also just copy that from the same source code. In the main method of Benchmark, remove the reference to SelectionSort.

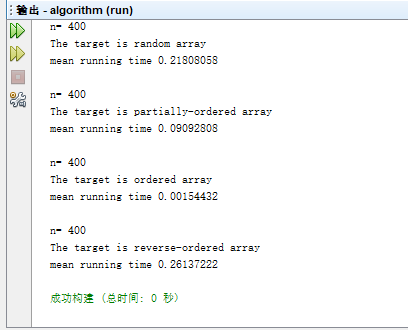
(Part 3) Measure the running times of this sort, using four different initial array ordering situations: random, ordered, partially-ordered and reverse-ordered. I suggest that your arrays to be sorted are of type Integer. Use the doubling method for choosing n and test for at least five values of n. Draw any conclusions from your observations regarding the order of growth.

* **Output** (few outputs to prove relationship)

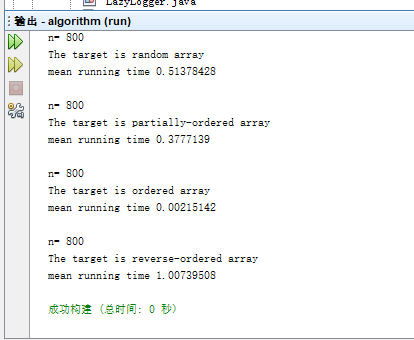
When the array has n members and n=200;



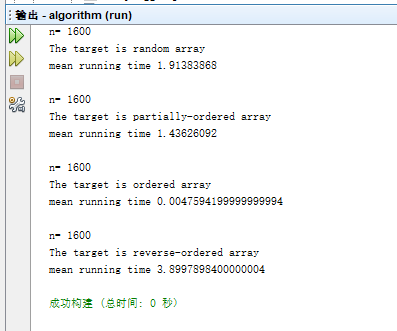
When the array has n members and n=400;



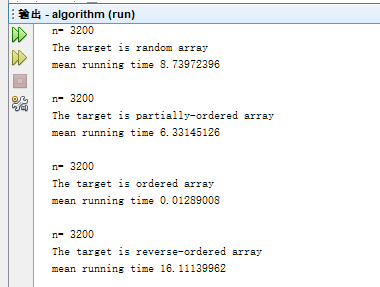
When the array has n members and n=800;



When the array has n members and n=1600;

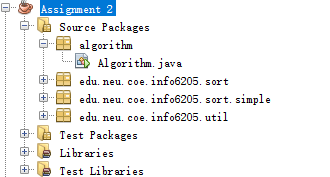


When the array has n members and n=3200;



* **Relationship conclusion**

I created a new main class called Algorithm to run this program.



When target array is ordered: Time is almost a constant number;

The Times of partially-ordered,random and reverse-ordered array all obey the graph of a function like F(n)=C (n^x )and C>0, X>0.

Besides,as the order of growth,the sort time of reverse-ordered array is greater than random array,

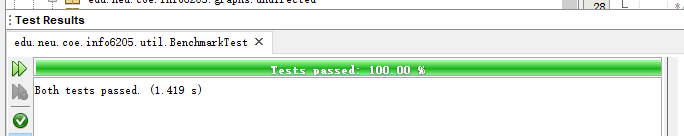
And the sort time of random array is greater than partially-ordered array.

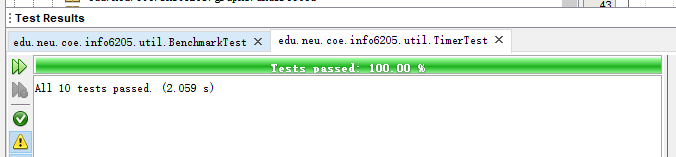
* **Evidence to support relationship** (screen shot and/or graph and/or spreadsheet)

Convert the output to a table as follow;



* **Screenshot of Unit test passing**

****

****