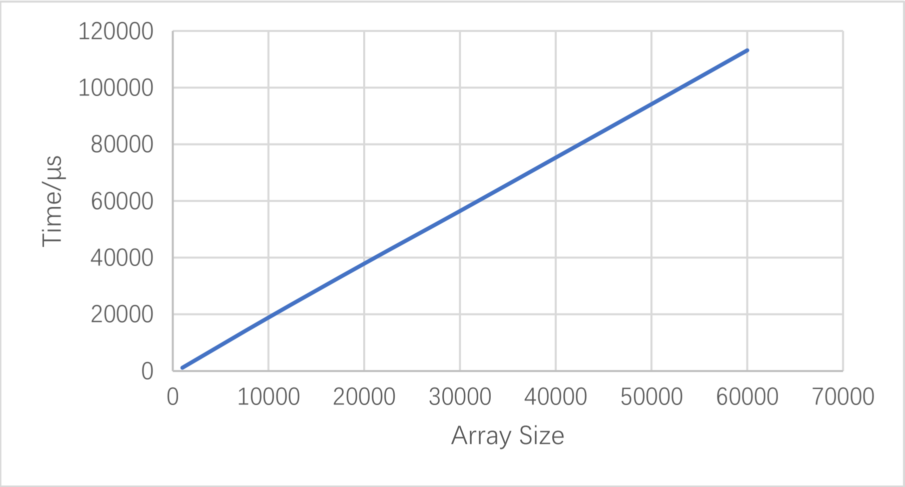
Exercises for 13 May

1. a. what values did you choose? Why these values?

I choose: 1e3, 1e4, 2e4, 3e4, 4e4, 5e4, 6e4. Because they're large enough to see the difference and prevent some noise.

b. Plot:



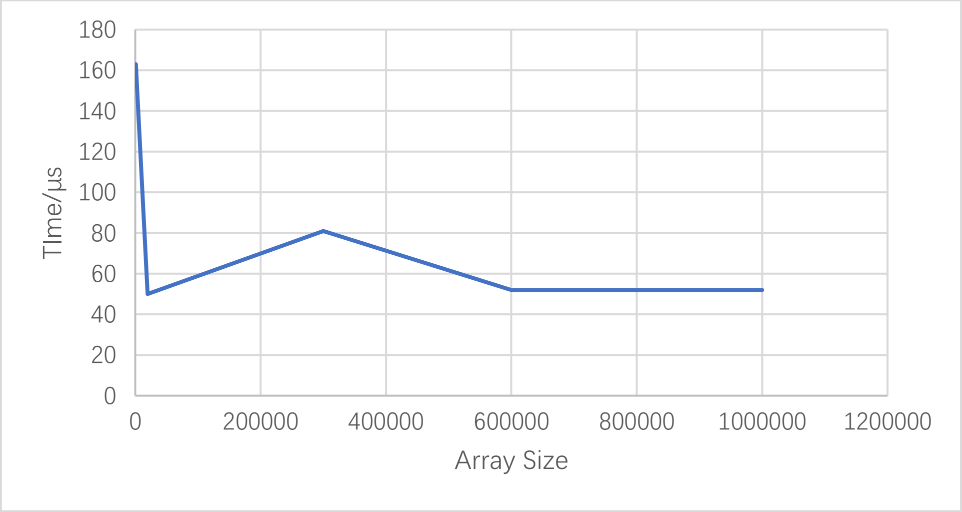
c. Does this conform to your expectations? Explain

I guess the relationship between size and time is linear. The result conforms to my expectation. From the code, we can see that to search for the largest number, every entry of the array is checked. Thus, when the size of array is big, time is approximately proportional to the size.

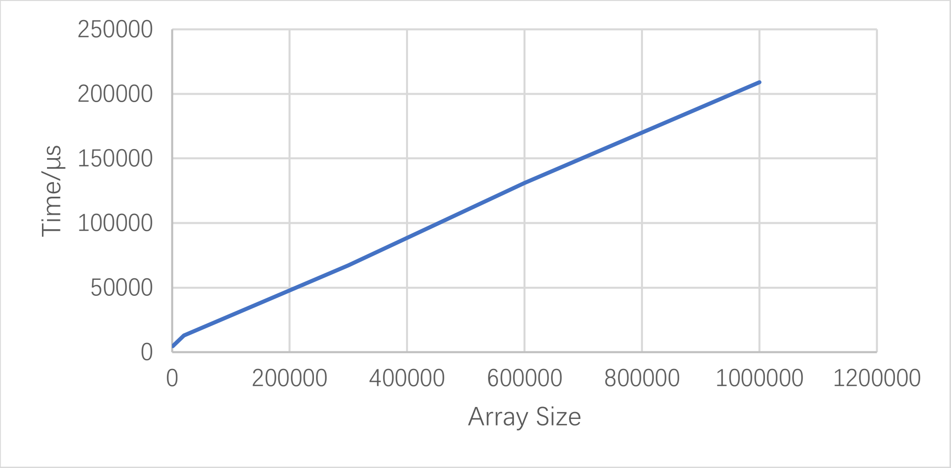
2. How long does it take to run for various working sizes?

It depends.

If inserting the value to the end of array, the time needed is short and almost constant, shown as following figure. There is some noise due to smallness.



If inserting the value to the front of array, the time needed is approximately proportional to the size, shown as following figure. Because every entry will be moved to insert a value at the start of array.



3. [optional] Re-implement IntArray using manual memory management (using new to allocate and either delete or smart pointers to clean up) and int pointers instead of std::vector<int>::[const_]iterator.
   Please see "IntArray.cpp" class NewArray.