After you have successfully completed your implementation of the min-heap operations and generated the tick count curve as described in the lab instructions, answer the following questions.

First, for each of the four methods below, describe briefly (in three sentences or less) how it works. In addition, answer the following questions: Did the method use any instance variables of the class, and if so, which ones? Did you create any helper methods, and if so, what do they do?

The four methods that you should describe are:

1.   `public Decreaser<T> insert(T thing){}`

2.   `void decrease(int loc){}`

3.   `public T extractMin(){}`

4.   `private void heapify(int where){}`

Now answer the following additional questions:

5. Include the tick-count graph obtained from the `HeapTimer` experiment (as described in the lab instructions) as a figure in your writeup. Does the curve have the shape suggested in the instructions?

6. What is the running time of calling `extractMin` n times? Give an asymptotic upper bound (O) on the running time as a function of n, and justify this bound.

7. Briefly describe what the heap property is and how it is maintained in the code during the `insert` and `extractMin` methods.

8. Could one side of a min-heap ever become very tall compared to the other side? To be precise, could the maximum height of any node in the left subtree of a min-heap ever differ from the max height of any node in the right subtree by more than (say) 5? If so, give a sequence of insertions that would produce such a min-heap; if not, explain why.

9. Name one advantage and one disadvantage of storing data in a min-first heap compared to storing it in an unordered list.