

1. `public Decreaser<T> insert(T thing)`
 - Parameters:
thing: this is an input for insert of type T. This input is what is being inserted into the heap.
 - Return value:
A new `Decreaser<T>` with the input (thing).
2. `void decrease(int loc)`
 - Parameters:
loc: input for the decrease method that takes a type int. This parameter gives a current location for the decrease method to start at.
 - Return value:
Does not have a return value.
3. `public T extractMin()`
 - Parameters:
No parameters
 - Return value:
This method takes the minimum value in the heap.
4. `private void heapify(int where)`
 - Parameters:
where: is a parameter of type int. Specifies where to start testing the rules of a binary tree by checking to see if the parent is larger than the child.
 - Return value:
no return value
5. Which instance variable in the `MinHeap` class holds the actual contents of the heap?
The `decreaser` holds to actual contents of the heap in the `MinHeap` class.
6. Which method(s) will call `heapify`?
The `extractMin` method will call `heapify` to resort the tree once the min has been taken out.
7. Why is the `decrease` method not private to the `MinHeap` classes?
The `MinHeap` class is not private maybe because it would need to be used in other classes to help run the code.