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

- Parameters:
  - thing: ‘thing’ parameter is what you want to push to the MinHeap class. The type is T for the template type. So you can add any type you want. The ‘thing’ will be created and added to MinHeap class.
- Return value:
  - Return ‘Decreaser<T>’ object which is just created and added into MinHeap class.

2. void decrease(int loc){}

- Parameters:
  - loc: ‘loc’ parameter is the position (or location) you want to start. Setting loc in this function will move (=bubble up) the specified node to the top of the Heap until it meets MinHeap requirements. ‘loc’ is used as the starting position for this function.
- Return value:
  - Void. There is no return value.

3. public T extractMin(){}

- Parameters:
  - No parameter
- Return value:
  - Return a T object. This function extracts the min(=top) node of the Heap. Since this is a MinHeap class so it will return the top node of the MinHeap class. And the function should take care of the Heap structure after extracting top node.

4. private void heapify(int where){}

- Parameters:
  - where: Parameter ‘where’ is the position you want to start the heapify. The ‘where’ parameter points the start node and the node is moved down(bubble-down) of the Heap if the value of the node is smaller than its children.
- Return value:
  - Void. There is no return value.

5. Which instance variable in the MinHeap class holds the actual contents of the heap?

- The answer is ‘private Decreaser<T>[] array’. I’ll add a comment about this. The ‘array’ variable just holds the address of a Decreaser<T> object. The ‘actual contents’ is inside the ‘Decreaser instance’ and it is in the ‘private T thing’ variable of the Decreaser class.

6. Which method(s) will call heapify?

- The answer is the ‘public T extractMin()’ function.

7. Why is the decrease method not private to the MinHeap class?

- If there is no modifier it means ‘Default’ modifier. ‘Default’ modifier can be accessed within the same class and the same package. The decrease method is in the MinHeap class, therefore it can be accessed by other methods in the same class or same package.