CS240, Spring 2022
Assignment 2: Question 3

Q3) Given an array $A[0 \ldots n-1]$ of numbers, show that if $A[i] \geq A[i-j]$ for all $j \geq \log(n)$, the array can be sorted in $O(n \log(\log(n)))$ time.
Hint: Partition A into contiguous blocks of size $\log(n)$; i.e. the first $\log(n)$ elements are in the first block, the next $\log(n)$ elements are in the second block, and so on. Then, establish a connection between the elements within two blocks that are separated by another block.

To start we will use the hint, we will partition the array into $n/log(n)$ blocks where each block has a size of $log(n)$. Since we are given that $A[i] \geq A[i-j]$ for all $j \geq \log(n)$, this means that each block's elements will be less then the elements to its right as they are a log(n) apart.

Moving forward we will use merge sort on each of the block, since there are $n/log(n)$ blocks where each one will take $O(\log(n) \log(\log(n)))$ (as merge sort takes $O(n \log(n))$ and the size of each block is $\log(n)$). Therefore the total time complexity will be:

$$O(\frac{n}{\log n} \log(n) \log(\log(n)))$$

Which simplifies to:

$$O(n \log(\log(n)))$$

There are a total of $\frac{n}{\log n}$ blocks that we will need to merge, the merge itself is constant we need to multiply it by the number of blocks we need to do, which gives us a time complexity of $O(\frac{n}{\log n})$. Thus our total time complexity is:

$$T(n) = O(n \log(\log(n))) + O(\frac{n}{\log n})$$

The overall time complexity will be determined by the term that grows the fastest so we will use the lime test to compare them:

$$\begin{aligned}
L &= \lim_{n->\infty} \frac{n \log(\log(n))}{\frac{n}{\log n}} \\
&= \lim_{n->\infty} \frac{\log(\log(n))}{\frac{1}{\log n}} \\
&= \lim_{n->\infty} \log(\log(n)) \log(n) \\
&= \infty
\end{aligned}$$

Therefore since $\frac{n}{\log n} \in o(n \log(\log(n)))$, it means that $n \log(\log(n))$ grows faster. It follows that our total time complexity will become:

$$T(n) = O(n \log(\log(n)))$$

Which is what we set up to prove.