4) The goal is to continuously deleteMin on the given heaps until there are no more records in the heaps.

Each of the sorted lists can be a representation of a heap

Pseudocode:

```
finalSortedArr[n]
minheap[k]
Populate minheap with the minumum values of each list. →(Insert is log(k) x (k times)
for i=0 to n:
    min = getMin(minheap) → O(1)
    next = min.left or min.right (which ever is closer in value to min) → O(1)
    finalSortedArr[i] = min → O(1)
    deleteMin(minheap) → O(log k)
    minheap.insert(next) → O(log k)
```

↳ This creates a runtime of $O(k \log k) + O(n(2(1) + 2(\log k)))$

↳ This is essentially $O(n \log k)$

This would only use $O(k)$ space, as minheap is the only thing that is stored in temporary memory