Next to each question, there is an indication of how hard we think each question is. Your answers should be as concise as possible while also fully explaining your solution. Please make an effort to write legibly.

The following questions are from the 2nd edition of the CLRS book. The text of the questions is placed here to avoid issues with different versions of the book.

6.2-6 (easy) Show that the worst-case running time of MAX-HEAPIFY on a heap of size $n$ is $\Omega(\log n)$. (Hint: For a heap with $n$ nodes, give node values that cause MAX-HEAPIFY to be called recursively at every node on a path from the root down to a leaf.)

6.4-4 (easy) Show that the worst-case running time of heapsort is $\Omega(n \log n)$.

6.4-5 (difficult) Show that when all elements are distinct, the best-case running time of heapsort is $\Omega(n \log n)$.

6.5-7 (easy) The operation \textsc{Heap-Delete}(A, i) deletes the item in node $i$ from heap $A$. Give an implementation of \textsc{Heap-Delete} that runs in $O(\log n)$ time for an $n$-element max-heap.

6.5-8 (medium) Give an $O(n \log k)$-time algorithm to merge $k$ sorted lists into one sorted list, where $n$ is the total number of elements in all the input lists. (Hint: Use a min-heap for $k$-way merging.)

7.2-6 (medium) Argue that for any constant $0 < \alpha \leq 1/2$, the probability is approximately $1 - 2\alpha$ that on a random input array, \textsc{Partition} produces a split more balanced than $1 - \alpha$ to $\alpha$.

7.4-6 (difficult) Consider modifying the \textsc{Partition} procedure by randomly picking three elements from array $A$ and partitioning about their median (the middle value of the three elements). Approximate the probability of getting at worst an $\alpha$-to-$(1 - \alpha)$ split, as a function of $\alpha$ in the range $0 < \alpha < 1$.

9.3-4 (difficult) Suppose that an algorithm uses only comparisons to find the $i^{th}$ smallest element in a set of $n$ elements. Show that it can also find the $i - 1$ smaller elements and the $n - i$ larger elements without performing any additional comparisons.