Algorithms: Solutions 3

Problem 1
Write an algorithm that combines Insertion-Sort and Merge-Sort.

The following algorithm calls Insertion-Sort for array segments whose length is at most k; the running time of this algorithm is $\Theta(n \cdot k + n \cdot \lg(n/k))$.

Insertion-Sort(A, p, r)
for $j \leftarrow p + 1$ **to** r
    do $key \leftarrow A[j]$
        $i \leftarrow j - 1$
        while $i \geq p$ **and** $A[i] > key$
            do $A[i+1] \leftarrow A[i]$
                $i \leftarrow i - 1$
        $A[i+1] \leftarrow key$

Combined-Sort(A, p, r, k)
if $r - p < k$
    then Insertion-Sort(A, p, r)
    else $q \leftarrow \lfloor (p + r)/2 \rfloor$
          Combined-Sort(A, p, q, k)
          Combined-Sort$(A, q + 1, r, k)$
          Merge(A, p, q, r)

Problem 2
Argue that the following algorithm correctly sorts the array $A[p..r]$:

Stooge-Sort(A, p, r)
1. **if** $A[p] > A[r]$
2.   **then** exchange $A[p] \leftrightarrow A[r]$
3. **if** $p + 1 \geq r$
4.   **then return**
5. $q \leftarrow \lfloor (r - p + 1)/3 \rfloor$
6. Stooge-Sort$(A, p, r - q)$ $\triangleright$ first two-thirds
7. Stooge-Sort$(A, p + q, r)$ $\triangleright$ last two-thirds
8. Stooge-Sort$(A, p, r - q)$ $\triangleright$ first two-thirds again

We prove the correctness of the algorithm by induction. Clearly, the algorithm works for one-element and two-element arrays, which provides the induction base. Now suppose that it works for all arrays shorter than $A[p..r]$ and let us show that it also works for $A[p..r]$.

After the execution of Line 6, $A[p..(r-q)]$ is sorted, which means that every element of $A[(p+q)..(r-q)]$ is no smaller than every element of $A[p..(p+q-1)]$; we write it as $A[(p+q)..(r-q)] \geq A[p..(p+q-1)]$. Thus, $A[(p+q)..r]$ has at least $\mathit{length}(A[(p+q)..(r-q)]) = r - p - 2q + 1$ elements each of which is no smaller than each element of $A[p..(p+q-1)]$.

After the execution of Line 7, $A[(p+q)..r]$ is sorted, which implies that

(1) $A[(r-q+1)..r]$ is sorted, and

(2) $A[(r-q+1)..r] \geq A[(p+q)..(r-q)]$.

Since $A[(p+q)..r]$ has at least $(r-p-2q+1)$ elements no smaller than each element of $A[p..(p+q-1)]$ and $\mathit{length}(A[(r-q+1)..r]) \leq r-p-2q+1$, we conclude that

(3) $A[(r-q+1)..r] \geq A[p..(p+q-1)]$.

Putting together (2) and (3), we conclude that

(4) $A[(r-q+1)..q] \geq A[p..(r-q)]$.

After the execution of Line 8, the array $A[p..(r-q)]$ is sorted. Putting this observation together with (1) and (4), we see that the whole array $A[p..r]$ is sorted.