Analysis of Algorithms: Solutions 3

```
                              X
                           X  X
                           X  X
                        X  X  X
                        X  X  X
                        X  X  X
                        X  X  X
number of               X  X  X
homeworks               X  X  X
               X     X  X  X  X
               X     X  X  X  X
               X     X  X  X  X
               X     X  X  X  X
               X     X  X  X  X
               -----------------
               5  6  7  8  9  10
                    grades
```

The histogram shows the distribution of grades, from 0 to 10.

Problem 1
Write pseudocode for the MERGE(A, p, q, r) procedure.

We use an auxiliary array $B[p..r]$, for storing the result of merging $A[p..q]$ and $A[q+1..r]$. After the merge, we copy the contents of $B[p..r]$ into $A[p..r]$. The running time is $\Theta(r-p+1)$.

MERGE(A, p, q, r)
$i \leftarrow p$ ▷ index in $A[p..q]$
$j \leftarrow q+1$ ▷ index in $A[q+1..r]$
$k \leftarrow p$ ▷ index in $B[p..r]$
while $i \leq q$ or $j \leq r$ ▷ merge $A[p..q]$ and $A[q+1..r]$
  do if $j > r$
      then $B[k] \leftarrow A[i]$
        $i \leftarrow i+1$
    else if $i > q$
      then $B[k] \leftarrow A[j]$
        $j \leftarrow j+1$
    else if $A[i] \leq A[j]$
      then $B[k] \leftarrow A[i]$
        $i \leftarrow i+1$
    else $B[k] \leftarrow A[j]$
        $j \leftarrow j+1$
    $k \leftarrow k+1$
for $k \leftarrow p$ **to** r ▷ copy the merged array to $A[p..r]$
  do $A[k] \leftarrow B[k]$

Problem 2

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 ← p + 1 to r
    do key ← A[j]
       i ← j − 1
       while i ≥ p and A[i] > key
           do A[i + 1] ← A[i]
              i ← i − 1
       A[i + 1] ← key
```

```
COMBINED-SORT(A, p, r, k)
if r − p < k
    then INSERTION-SORT(A, p, r)
    else q ← ⌊(p+r)/2⌋
         COMBINED-SORT(A, p, q, k)
         COMBINED-SORT(A, q + 1, r, k)
         MERGE(A, p, q, r)
```