Analysis of Algorithms: Solutions 8

```
                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
----------------------
3  4  5  6  7  8  9  10
        grades
```

Problem 1
Write an efficient algorithm for scheduling several lectures, using as few lecture halls as possible. For each lecture i, we know its start time s_i and finish time f_i.

Greedy-Hall-Selector(s, f, n)
sort lectures in the ascending order of their finish times
assigned $\leftarrow 0$ $\quad\triangleright$ number of lectures already assigned to specific halls
halls $\leftarrow 0$ $\quad\triangleright$ number of lecture halls
for $i \leftarrow 1$ **to** n
  do $hall_i \leftarrow 0$ $\quad\triangleright$ hall for the ith lecture; 0 means "no specific hall yet"
while *assigned* $< n$
  do *halls* $\leftarrow$ *halls* $+ 1$
    finish $\leftarrow -\infty$
    for $i \leftarrow 1$ **to** n $\quad\triangleright$ in sorted order
      do if $hall_i = 0$ and $s_i \geq$ *finish*
        then $hall_i \leftarrow$ *halls*
          finish $\leftarrow f_i$
          assigned $\leftarrow$ *assigned* $+ 1$
return $hall_1..hall_n$ $\quad\triangleright$ array of hall assignments

The time complexity of this algorithm is $O(n \cdot halls)$, where n is the number of lectures and *halls* is the final number of halls.

Problem 2

Imagine that you need to pay n cents, using the smallest possible number of coins. You have an unlimited supply of quarters, dimes, nickels, and pennies. Give an efficient algorithm that finds the minimal set of coins for a given amount n.

FEWEST-COINS(n)
$quarters \leftarrow \lfloor \frac{n}{25} \rfloor$
$n \leftarrow n - 25 \cdot quarters$
$dimes \leftarrow \lfloor \frac{n}{10} \rfloor$
$n \leftarrow n - 10 \cdot dimes$
$nickels \leftarrow \lfloor \frac{n}{5} \rfloor$
$n \leftarrow n - 5 \cdot nickels$
$pennies \leftarrow n$
return *quarters*, *dimes*, *nickels*, *pennies*

Problem 3

Using Figure 17.4(b) in the textbook as a model, draw an optimal-code tree for the following set of characters and their frequencies:

a:4 b:6 c:10 d:12 e:18 f:40 g:50 h:60