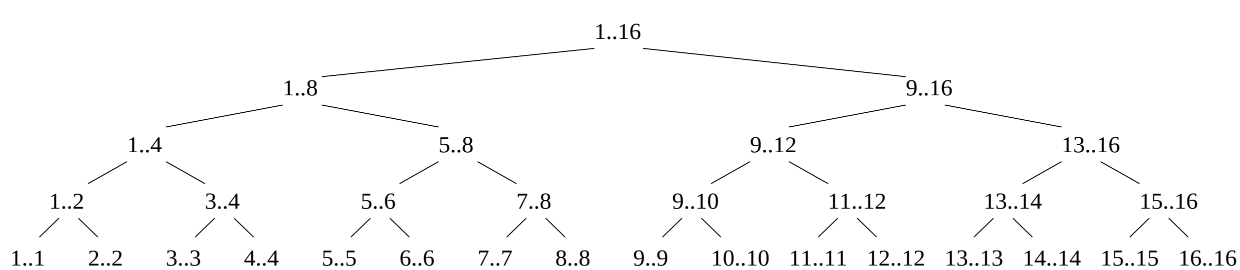
Analysis of Algorithms: Solutions 10

```
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
                              X
number of                     X
homeworks                     X
                              X
                              X
               X  X           X
               X  X     X  X  X
            X  X  X     X  X  X
            --------------------
            4  5  6  7  8  9  10
                   grades
```

Problem 1
Draw the recursion tree for the MERGE-SORT procedure on a sixteen-element array. Explain why dynamic programming is ineffective for speeding up MERGE-SORT.



The MERGE-SORT procedure does *not* have overlapping subproblems, that is, all nodes of the recursion tree are distinct. We cannot re-use the results of recursive calls; hence, dynamic programming does not improve the efficiency.

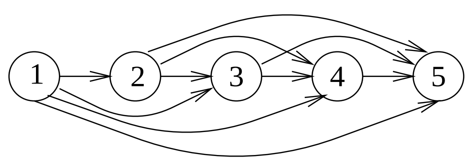
Problem 2
Suppose that you drive along some road, and you need to reach its end. Initially, you have a full tank, which holds enough gas to cover a certain distance d. The road has n gas stations, where you can refill your tank. The distances between gas stations are represented by an array $A[1..n]$, and the last gas station is located at the end of the road. You wish to make as few stops as possible. Give an algorithm CHOOSE-STOPS(d, A, n) that identifies all places where you have to refuel.

```
CHOOSE-STOPS(d, A, n)
stations ← ∅        ▷ set of selected gas stations
d-left ← d          ▷ distance that corresponds to the remaining gas
for i ← 1 to n
    do if d-left < A[i]      ▷ cannot reach the next gas station? then refuel
          then stations ← stations ∪ {i − 1}
               d-left ← d
       d-left ← d-left − A[i]      ▷ drive to the next station
return stations
```

The algorithm runs in linear time, that is, its complexity is $\Theta(n)$.

Problem 3
What is the maximal possible number of edges in a directed acyclic graph with V vertices?

The maximal number of edges is $\frac{V\cdot(V-1)}{2}$. To construct a graph with that many edges, we enumerate its vertices from 1 to V, and put an edge from every vertex to every higher-number vertex. That is, the graph includes an edge (i, j) if and only if $i < j$. For example, if $V = 5$, then the graph is as follows:



To prove that this number is maximal, we observe that, for every two vertices i and j, an acyclic graph may include either the edge (i, j) or (j, i), but not both. Thus, the total number of edges is no greater than the number of vertex pairs, which is $\frac{V\cdot(V-1)}{2}$.