b.

The recursive procedure forms a tree with branching factor 4 (4 possible activity), with each branch created by either full service, regular service, minor service and no service.

From the root, we start with 1 node, then on the next level, we will have 4 nodes, and then the next level, will be 16 nodes, and so on. With couple more levels, we can summarize there is a geometric series:

To calculate the total number of nodes in the tree after k depth (levels), we sum the geometric series.

This give us an asymptotic lower bound Ω(4k) on the worse case time complexity.

d.

From the procedure of implementation of building the bottom-up table below:

for (int i(current hour) = k – 1(last hour); i >= 0; i--) {

for (int h(hours since service) = 0; h <= i + 1; h++) {

// add to table

}

}

We can write the two for loop into mathematic form which is equivalent to

his give us an asymptotic upper-bound O(k2) on the worse case time complexity.