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.

$$\sum_{i=0}^{k} 4^{i} = 1 + 4 + 4^{2} + 4^{3} + \ldots + 4^{k} = \frac{4^{k+1} - 1}{4 - 1} = \frac{4^{k+1} - 1}{3}$$

This give us an asymptotic lower bound $\Omega(4^k)$ 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

$$\sum_{i=0}^{k-1}\sum_{h=0}^{i} 1 = \sum_{i=1}^{k}\sum_{h=1}^{i+1} 1 = \sum_{i=1}^{k}(i+1)$$

$$\sum_{i=1}^{k} i + \sum_{i=1}^{k} 1 = (\sum_{i=1}^{k} i) + k$$

$$\frac{k(k+1)}{2} + k = \frac{k^{2} + k}{2} + k = \frac{k^{2} + 3k}{2}$$

his give us an asymptotic upper-bound $O(k^2)$ on the worse case time complexity.