1) What is the asymptotic runtime of the algorithm shown below?

```
for i from 0 to n-1
    “Line 2”
    for j from 0 to n-1
        “Line 4”
        for k from 0 to n-1
            “Line 6”
```

$O(n^2)$

2) Call your answer to the previous question $g(n)$. Justify your answer to the previous question by finding the constant multiple and point that it starts to apply: (Fill in the boxes; show and supporting work or derivation below)

$$f(n) \leq 2 \cdot g(n) \text{ whenever } n \geq 1$$

The runtime is $n(n + n) = n^2 + n^2 \leq 2n^2$.

3) In the pseudocode below, “Search_Database(m)” is the function of interest, and runs in $O(h(m))$ time. Everything else is trivial. What is the asymptotic growth rate of this algorithm? (Bonus Question) (Technically to simplify this like I do below, we need to know that $h(m)$ is an increasing function. But this is a safe assumption because it measures the runtime of an algorithm.)

```
for i from 0 to n-1
    “Search_Database(i)”
    for j from 0 to n-1
        “Return_Records(j)”
        for k from i to j
            “Do_stuff(i,j,k)”
```

First note that we say that everything other than “Search_Database(m)” is trivial, so we can safely ignore lines 4 and 6. Now “Search_Database(m)” runs $n$ times with inputs ranging from 0 to $n - 1$. Hence we have the runtime $O(n \cdot h(n))$ as derived below.

Plugging in each $i$ we get a runtime of $O(h(1)) + O(h(2)) + \cdots + O(h(n - 1))$.

Now note that $O(h(i))$ is $O(h(n))$ because $i \leq n$ so $h(i) \leq h(n)$. Hence we get:

$$O(h(n)) + O(h(n)) + \cdots + O(h(n)) = O(n \cdot h(n))$$