5. (2-2) What is the value returned by the following function? What is the runtime in big-O?

```c
int pesky(int n) {
    int r = 1;
    for (int i = 1; i <= n; ++i)
        for (int j = 1; j <= i; ++j)
            for (int k = j; k <= j+i; ++k)
                r += 1;
    return r;
}
```

The innermost loop (indexed by \( k \)) increases \( r \) by 1 a total of \( i \) times, so the loop could be replaced by \( r += i; \). Then the second loop (indexed by \( j \)) increases \( r \) by \( i \) a total of \( i \) times, so can be replaced by \( r += i^2; \). Finally the outer loop (indexed by \( i \)) increases \( r \) by \( i^2 \) for \( i \) running from 1 to \( n \). [Remark: For the first time the body of the loop depends on the index of the loop.] This is problem 1-11 which we solved in class using induction. The loop computes \( \sum_{i=1}^{n} i^2 = n(n + 1)(2n + 1)/6 \), which is \( O(n^3) \).

The runtime has the same \( O(n^3) \) complexity since the total number of steps is proportional to the number of times 1 is added to \( r \). To be pedantic about it, the loop index \( k \) is compared and incremented for each increment of \( r \), thus directly in proportion. The other loop index steps occur less often. QED.

7. Show \( n^2 \) is \( O(2^n) \).

We all sort of know that \( 2^n \) grows much faster than \( n^2 \) (exponential vs quadratic growth). How can we show it using only elementary algebra?

I suggest using proof by induction (but depending on the \( n/2 \) case, not the \( n - 1 \) case in the inductive step).

I’ll show that \( n^2 \leq 2^n \) (ie. I’ll use \( C = 1 \)).

Base cases:

<table>
<thead>
<tr>
<th>( n )</th>
<th>( n^2 )</th>
<th>( 2^n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>49</td>
<td>128</td>
</tr>
</tbody>
</table>

We see that the assertion is true for \( n = 4 \) through 7. My plan is that the inductive step for \( n \) will be based on a previous case of \( n/2 \) or \( (n + 1)/2 \), whichever is an integer. Thus we’ll have a basis for induction starting at \( n = 8 \) and using the inequality starting at \( k = 4 \).

Inductive step: Suppose \( n = 2k \) is even. Then
\[
n^2 = 4 \times k^2 \leq 4 \times 2^k \quad \text{(using inductive hyp.)}
\]
\[
= 2^{k+2}
\]
\[
< 2^{2k}, \text{ if } k \geq 2. \text{ Done with the even } n \text{ case.}
\]

Now suppose \( n \) is odd, \( n = 2k - 1 \).
\[
n^2 = 4k^2 - 4k + 1 \leq 4 \times 2^k - 4k + 1 \quad \text{(by inductive hyp.)}
\]
\[
< 2^{k+2} \quad \text{(using that } 4k-1 \text{ is pos for pos integer } k.)
\]
\[
< 2^{2k-1}, \text{ if } k \geq 3. \text{ Done with the odd } n \text{ case.}
\]
QED.