2) (10 pts) ANL (Algorithm Analysis)

An algorithm that processes a list of size n takes $O(\sqrt{n}lgn)$ time. On Shannon's computer, when she runs the algorithm on a list of size $\mathbf{n = 2^{16}}$**,** her computer takes **c** milliseconds. (Shannon is very secretive, so she hasn't told you the value of **c** unfortunately!) **In terms of c,** how long, **in milliseconds,** should we expect the algorithm to take on her computer when she is processing a list of size $\mathbf{2^{20}}$? (Your answer should be of the form kc, where k is a positive real number.)

Let T(n) = $d\sqrt{n}lgn$, be the run time of the algorithm on a list of size n, where d is some constant, different than the variable c mentioned in the problem. Using the given information we have:

$$c = T(2^{16}) = d\left(\sqrt{2^{16}}\right)(\lg 2^{16}) = (2^8)(16lg2)d = (2^{12}lg2)d$$

We seek the value of T(2^{20}):

$$T(2^{20}) = d\left(\sqrt{2^{20}}\right)(\lg 2^{20}) = (2^{10})(20lg2)d = 5(2^{12}lg2)d = \mathbf{5c}$$

It follows that the desired answer is 5c. (Shannon should expect this list to take about five times longer than the previous list.)

Note: Notice that the value of d cancels out, so we can simply calculate the ratio of T(2^{20})/T(2^{16}) and this will reveal the value of 5 as the multiplicative factor.

Grading: 1 pt for plugging in 2^{16} into the Big-Oh function,
3 pts for simplifying it to $2^{12}lg2$, or another suitable form.
1 pt for plugging in 2^{20} into the Big-Oh function,
3 pts for simplifying it to $(2^{10})(20lg2)$, or another suitable form.
2 pts for doing substitution or division or anything equivalent to arrive at the answer 5c.