Seth Jacobs
Asymptotics drill

1) When doubling n in an algorithm which has a run time of n^2 the algorithm gets slower by a factor of 4 each time you double n. If you run the algorithm after adding one to n, each time it is slower by adding (n*2) +1 to the total amount of operations.

2) When doubling n in an algorithm which has a run time of n^3 the algorithm gets slower by a factor of 8 each time you double n If you run the algorithm after adding one to n, each time it is slower by a factor $3n^2$ +3n+1

3) When doubling n in an algorithm which has a run time of $100n^2$ the algorithm gets slower by a factor of 4 each time you double n. If you run the algorithm after adding one to n, each time it is slower by a factor 200n+100

4) When doubling n in an algorithm which has a run time of n log n the algorithm gets slower by a factor of a little more than 2.5 each time you double n. If you run the algorithm after adding one to n, each time it is slower by adding 4 to the amount of total operations.

5) When doubling n in an algorithm which has a run time of 2^n the algorithm gets slower by a factor of 2^n each time you double n. If you run the algorithm after adding one to n, each time it is slower by a factor of 2.

If you have a computer that can run 10^{10} operations per second, the largest input for an algorithm that has a run time of

1) n^2 is 6,000,000
2) n^3 is 33,019
3) $100n^2$ is 600,000
4) n log n is 1,290,000,000,000
5) 2^n is 45
6) 2^{2n} Is 22