Exercise 2: This would be if all the data points lie on the outside making them all hull points

Exercise 3: If the array was sorted this can improve the performance of our because we would only need to know the max and min

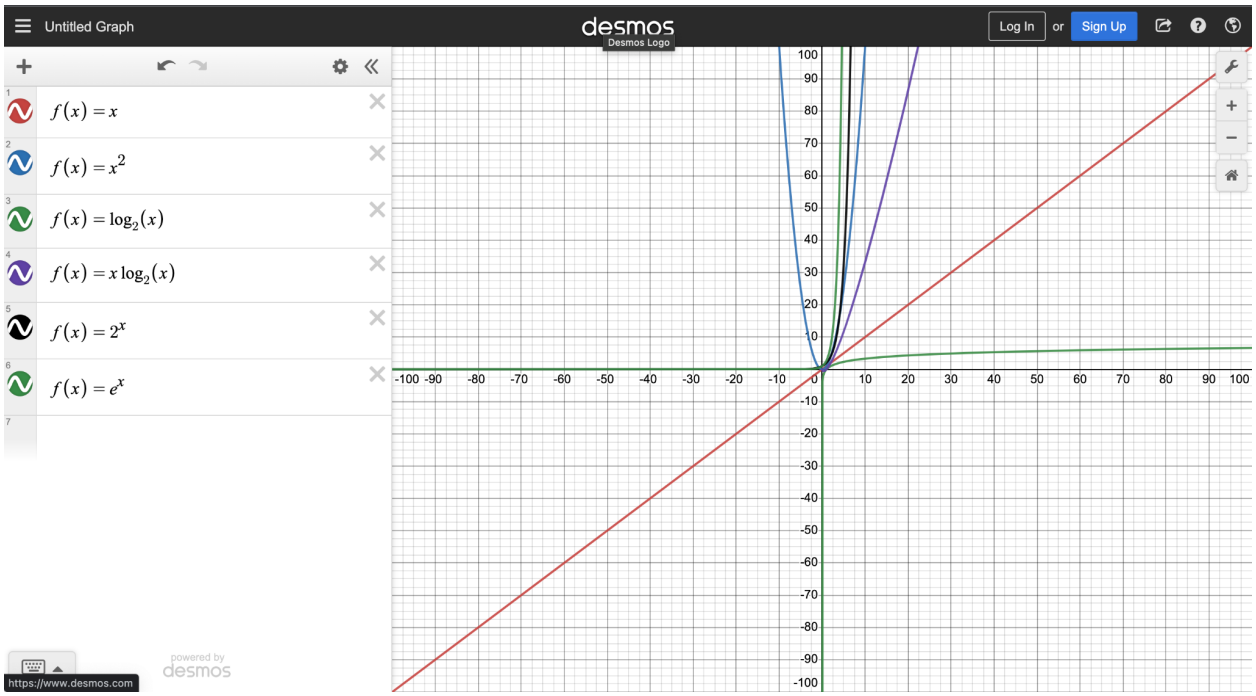
Exercise 4: Algorithm 3 is 2n and algorithm 2 is not known because we don't know what the sort algorithm looks like

Exercise 5: the number of times the distance is computed for our algorithm is n(n-1)/2

Exercise 6: The result is a triangle of X's

X X X X X X X
X X X X X X
X X X X X
X X X X
X X X
X X
X

Exercise 7:



Exercise 8: loga(xy)=loga(x)+loga(y)
By exponential rules then $a^{(\log_a(x)+\log_a(y))} = xy$
$a^{(\log_a(x)+\log_a(y))} = a^{(\log_a(x))}a^{(\log_a(y))}$
We simplify the right side to $a^{(\log_a(xy))}$
Making both equivalent

Exercise 9:
They cross at about x = 35
Yes but it is not clean numbers though I needed to use the quadratic formula

Exercise 10: Yes they both have the same leading degree so that means they essentially will grow at the same rate.

Exercise 11: c would be any integer greater than 9

Exercise: 12: c could be 2000

Exercise 13: From the graph it looks like for all N = 200 up until about 1000

Exercise 14:
$1000g(n) = 1000n^2$
$1000g(n) = 9n^2 + 991n^2$
$9n^2 + 991n^2 >= 9n^2+1000n$
$991n^2 = 1000n$
So when n >= approximately 1

Exercise 15:
Because we know that this case is true when n > 1, so when n is positive we will be able to use this when N is any positive integer.

Exercise 16:
If f(n)+f(n) = the max O(g(n)g(n))
It implies that out of the 2 g(n) they would pick the one with the largest power

Exercise 17:

The smallest k that I tested would be 1/2
The number of comparisons do match
Yes the input data affects the number of comparisons

Exercise 19:

- Metropolis Algorithm for Monte Carlo

- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method