1. With 3 nested loops, the time complexity should be O(n^3). With an n of 3, r is 20. With an n of 4, r is 40. With an n of 5, r is 70.

2.

Show directly that $f(n) = n^2 + 3n^3 \in \Theta(n^3)$

i) CLAIM: $n^2 + 3n^3 \in O(n^3)$

Proof: by O definition, $f(n)$ is $O(n^3)$ iff $f(n) \leq c \cdot n^3$ for some real numbers $n > k = 1$ and c. Let $c = 4$ and $k = 2$, $(2)^2 + 3(2)^3 \leq 4(2)^3$

$$4 + 24 \leq 32$$

$$28 \leq 32 \checkmark$$

$\exists c$ such that $f(n) \leq c \cdot n^3$, so $f(n) \in O(n^3)$.

ii) CLAIM: $n^2 + 3n^3 \in \Omega(n^3)$

Proof: By Ω definition, $f(n)$ is $\Omega(n^3)$ iff $f(n) \geq c \cdot n^3$ for some real numbers c and k, where $n \geq k$. Let $c = 3$ and $k = 2$. $(2)^2 + 3(2)^3 \geq 3(2)^2$

$$4 + 24 \geq 24$$

$$28 \geq 24 \checkmark$$

$\exists c$ such that $f(n) \geq c \cdot n^3$, so $f(n) \in \Omega(n^3)$.

iii) CLAIM: $f(n) = n^2 + 3n^3 \in \Theta(n^3)$.

Given $f(n) = n^2 + 3n^3 \in O(n^3) \wedge f(n) = n^2 + 3n^3 \in \Omega(n^3)$, it follows that $f(n) = n^2 + 3n^3 \in \Theta(n^3)$ by the definition of Θ.

3.

$2^{n+1} \to \Theta(2^n)$

substitute constant a, arbitrarily chosen (but is a positive integer)

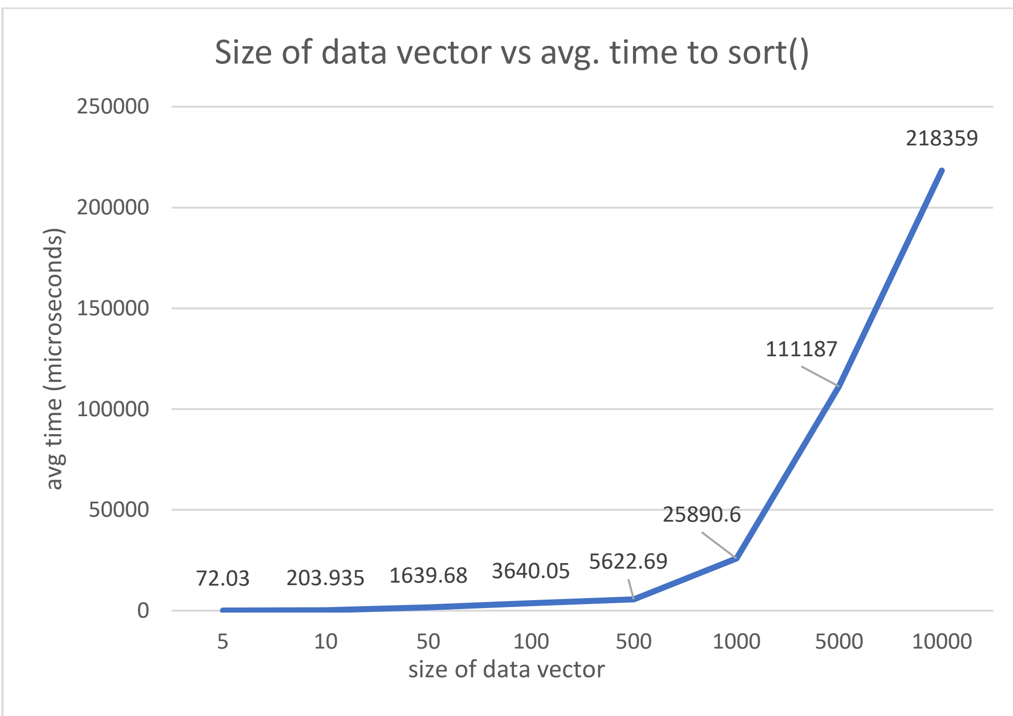
O. Yes, by defn. of exponents, a^{n+1} is equal to $a \cdot a^n$, so the inequality $a \cdot a^n \le c \cdot a^n$ for any $c > a$. So, $\exists c$ s.t $\forall a\,(a^{n+1} \geq c \cdot a^n)$

Ω. Yes, Again by defn. of exponents, a^{n+1} is equal to $a \cdot a^n$, so the inequality $a \cdot a^n \ge c \cdot a^n$ for any $c < a$.
So, $\exists c$ s.t $\forall a\,(a^{n+1} \ge c \cdot a^n)$

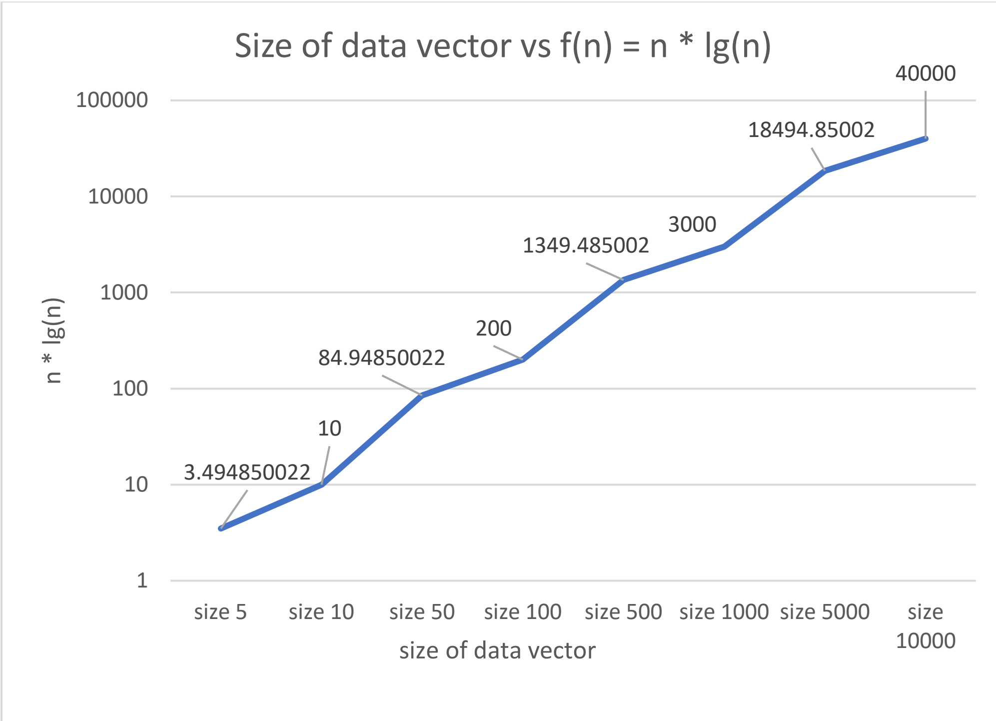
Θ. $2^{n+1} \in O(2^n) \wedge 2^{n+1} \in \Omega(2^n) \longrightarrow 2^{n+1} \in \Theta(2^n)$

4. For the worst case, it is necessary to check every possible edge, meaning each spot in the matrix needs to be accessed. This means the worst case is O(n^2).

5. Average sort() times of 200 trials of each size.



f(n) = n * lg(n) of sort() for 200 trials of each size.



I was a little worried about the state of my log graph before finally remembering I needed to actually change the axis to logarithmic in Excel. With that done, the growth rate looks about right for n lg n as shown in the Foundations textbook. The cppreference sort() page and the Introsort Wikipedia page linked therein state that sort() usually uses introsort, which is a hybrid algorithm. Introsort uses quicksort, then heapsort, and then insertion sort as needed. This keeps practical performance near that of quicksort and worst-case performance near heapsort. Presumably this is what keeps the complexity low enough for it to be a widely useful function for the standard library.