1.(a)

$\frac{n(n-1)}{2}$ is $O(n^2)$

If we can find constants m and k such that

$k \times n^2 > \frac{n(n-1)}{2}$ for all $n \geq m$ then

the algorithm is $O(n^2)$

Find values of k and m so that this is true

$k = 1$, and $m = -1$

then $n^2 \geq \frac{n(n-1)}{2}$ for all $n \geq -1$

(b) $\max(n^3, 10n^2)$ is $O(n^3)$

① when $n < 10$, $n^3 > 10n^2$

If we can find constants m and k such that:

$k \cdot n^3 > n^3$ for all $n \geq m$ then

the algorithm is $O(n^2)$

$k = 2$, and $m = 0$

then $2n^3 \geq n^3$ for all $0 \leq n < 10$

② when $n \geq 10$, $n^3 < 10n^2$

If we can find constants m and k such that:

~~$k \times 10n^2$~~

$k \cdot n^3 > 10n^2$ for all $n \geq m$ then

the algorithm is $O(n^3)$

$k = 1$, and $m = 10$

then $n^3 \geq 10n^2$ for all $n \geq 10$

(c) $\sum_{i=1}^{n} i^k$ is $O(n^{k+1})$ for integer k

$$n^{k+1} = n^k + n^k + \cdots + n^k$$

$$\sum_{i=1}^{n} i^k = 1^k + 2^k + \cdots + (n-1)^k + n^k$$

$$\lim_{n\to\infty} \frac{1^k + 2^k + \cdots + n^k}{n^k + n^k + \cdots + n^k} \leq \frac{n^k + n^k + \cdots + n^k}{n^k + n^k + \cdots + n^k} = 1$$

then $\sum_{i=1}^{n} i^k$ is $O(n^{k+1})$ for integer k

(d) If $p(x)$ is any k^{th} degree polynomial with a positive leading coefficient, then $p(n)$ is $O(n^k)$

$$p(x) = \sum_{n=0}^{k} a_n x^n, \qquad a_k \neq 0$$

$$\lim_{n\to\infty} \frac{p(x)}{n^k} = a_1 + a_2\left(\frac{1}{n}\right) + \cdots + a_k\left(\frac{1}{n^k}\right)$$

$$= a_1$$

then $p(n)$ is $O(n^k)$

2.

(a) $n^{\log n}$; $(\log n)^n$

$n^{\log n} = e^{(\log n)\cdot(\log n)}$

$(\log n)^n = e^{n \cdot \log(\log n)}$

$f(x) = (\log x)^2 \cdot g(x) = x \cdot \log(\log x)$

$\lim_{x \to \infty} \frac{(\log x)^2}{x \cdot \log(\log x)}$

$= \frac{2(\log x)\frac{1}{x}}{\log(\log x) + \frac{x}{x \log x}}$

$= \frac{2(\log x)^2}{x \log x(\log(\log x)) + x}$

$= 0$

then when $x \to \infty$, $n^{\log n} < (\log n)^n$

so $(\log n)^n$ grows faster

(b) $\log n^k$; $(\log n)^k$

$\log n^k = k \log n$

$kx < x^k \quad (x > 0, k > 0)$

$\Rightarrow (\log n)^k$ grows faster

(c) $n^{\log\log\log n}$; $(\log n)!$

$n! \geq e n^n [n+\frac{1}{2}] \cdot e^{-n}$

$(\log n)! \geq e (\log n)^n \cdot [\log n + \frac{1}{2}] \cdot e^{-\log n}$

$(\log n)^n \cdot \log n = e^n [\log n [\cdot \log\log(n)]]$

$\Rightarrow (\log n)^n \cdot \log n \, e^{-\log n} = e^n [\log n \cdot \log\log(n) - \log n]$

$= n^{(\log\log n - 1)}$

$n^{(\log\log n - 1)} > n^{(\log\log\log n)}$ $(x - 1 > \log x)$

$\Rightarrow (\log n)!$ grows faster

(d) n^n ; $n!$

$\frac{n!}{n^n} \geq 0$ $(n! > 0, n^n > 0)$

$\lim_{n\to\infty} \frac{n!}{n^n} = \frac{n \cdot (n-1) \cdot \cdots 1}{n \cdot n \cdot \cdots n} = 0$

$\Rightarrow n^n$ is faster

3. If $f_1(n)$ is $O(g_1(n))$ and $f_2(n)$ is $O(g_2(n))$ where f_1 and f_2 are positive functions of n, show that the function $f_1(n) + f_2(n)$ is $O(\max(g_1(n), g_2(n)))$

$$\lim_{n\to\infty} \frac{g_1(n)}{g_2(n)} = \begin{cases} +\infty \\ 0 \\ C, (C \neq 0) \end{cases}$$

if $\lim_{n\to\infty} \frac{g_1(n)}{g_2(n)} = +\infty$,

$\lim_{n\to\infty} \frac{g_2(n)}{g_1(n)} = 0$

$\Rightarrow O(\max(g_1(n), g_2(n))) = O(g_1(n) + g_2(n))$

$f_1(n) \leq C_1 g_1(n)$

$f_2(n) \leq C_2 g_2(n)$

$f_1(n) + f_2(n) \leq C_1 g_1(n) + C_2 g_2(n)$

$\leq (g_1(n) + g_2(n)) \cdot (C_1 + C_2)$

$\Rightarrow f_1(n) + f_2(n) = O(g_1(n) + g_2(n)) = O(\max(g_1(n), g_2(n)))$

4. Prove or disprove: Any positive n is $O(\frac{n}{2})$
If we can find constants m and k such that
$k \cdot \frac{n}{2} > n$ for all $n \geq m$. then
the algorithm is $O(n^2)$
$k = 3$, and $m = 0$
then $\frac{3}{2}n > n$, for all $n > 0$.

5. Prove or disprove 3^n is $O(2^n)$
If we can find constants m and k such that
$k \cdot 2^n > 3^n$ for all $n \geq m$. then
the algorithm is $O(n^2)$
$\Rightarrow k > \frac{3^n}{2^n}$, $\lim_{n \to \infty} \frac{3^n}{2^n} = +\infty$
so can not find a k
$\Rightarrow 3^n$ is not $O(2^n)$