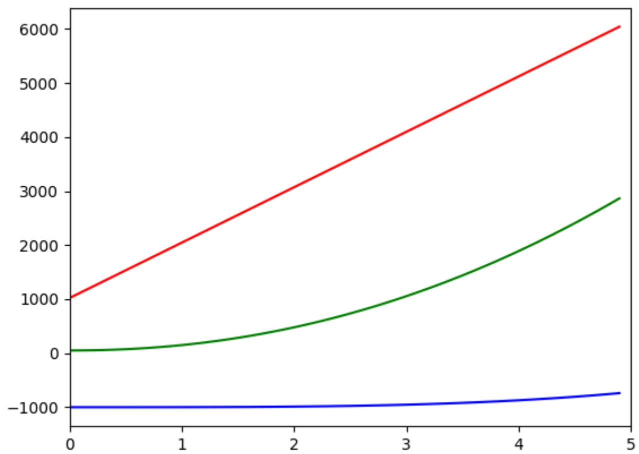
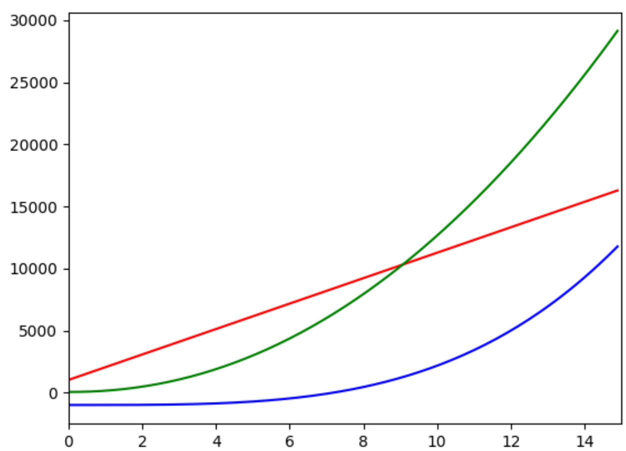
Q1:

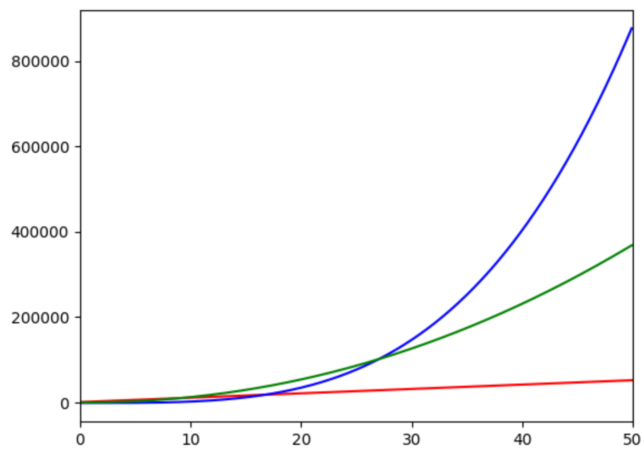
As n starts off and is small, f1 has the greatest value, followed by f3 and finally f2.



As n continues to increase, f3 overtakes f1 in value, while f2 is the least but showing exponential growth



As n is very large, f2 has the largest value, and f3 has the second highest, while f1 is the least. f3 and f1 have lower rates of growth than f2.



Python Code:

```
import math
import numpy as np
import matplotlib.pyplot as plt

msize = [5, 15, 50]

# red dashes, blue squares and green triangles
# plt.plot(t, t, 'r--', t, t**3.5 - 2**10, 'bs', t, 100*t**2.1 + 50, 'g^')
for item in msize:
    t = np.arange(0, item, 0.1)
    plt.plot(t, (2**10)*t + (2**10) , 'red', t, t**3.5 - 1000, 'blue', t,
100*t**2.1 + 50, 'green')
    plt.xlim(0, item)
    plt.rcParams["figure.figsize"] = (7,7)
    plt.show()
```

Q2:

$2^{(n+1.3)} \le c \cdot 2^n$
$2^{1.3} \cdot 2^n \le c \cdot 2^n$

$n_0 = 1 \quad c = 3$

$f(n) = O(2^n)$ since for
$c = 3$, $3(2^n) > f(n)$ when $n > 1$

$3^{(2 \cdot n)} \le c \cdot 3^n$
$3^2 \cdot 3^n \le c \cdot 3^n$

$n_0 = 1 \quad c = 10$

$f(n) = O(3^n)$ since
$c = 10$, $10(3^n) > f(n)$ when $n > 1$

Q3:

3. a. $(4n)^{150} + (2n + 1024)^{400} \le c(20n^{400} + (n + 1024)^{200})$

If $n_0 = 1$

$(4^{150}) + (2(1) + 1024)^{400} \le c(20(1)^{400} + (1 + 1024)^{200})$

$4^{150} + 1026^{400} \le c(20^{400} + 1025^{200})$

$\frac{4^{150} + 1026^{400}}{20^{400} + 1025} \le c$

$n_0 = 1 \quad c = \frac{4^{150} + 1026^{400}}{20^{400} + 1025} + 1$

$f(n) = O(g(n))$ since when $c = \frac{4^{150} + 1026^{400}}{20^{400} + 1025} + 1$, $c \cdot g(n) > f(n)$ when $n > 1$

b. $n^{1.4} \cdot 4^n \le c(n^{200} + 3.99)^n$

If $n_0 = 1$

$(1)^{1.4} \cdot 4^{(1)} \le c(1^{200} + 3.99)^1$

$4 \le c(4.99)$

$n_0 = 1 \quad c = 1$

$f(n) = O(g(n))$ since when $c = 1$, $c \cdot g(n) > f(n)$ when $n > 1$

c. $2^{\log(n)} \le c\, n^{1024}$

If $n_0 = 1$

$2^{\log(1)} \le c \cdot (1)^{1024}$

$1 \le c$

$n_0 = 1 \quad c = 2$

$f(n) = O(g(n))$ since when $c = 2$, $c \cdot g(n) > f(n)$ when $n > 1$

Q4:

4.

$i = 1$

while $i \le n$ **do** n ← Iterates n times since $i = i+1$

$A[i] = i$ n

$i = i + 1$ n

end while

for $j \leftarrow 1$ to n **do** n ← For loop will run n times

$i = j$ n

while $i \le n$ **do** $n\log(n)$

$A[i] = i$ $n\log(n)$

$i = i + j$ $n\log(n)$

end while

end for

As the for loop progresses, the while loop will under go less and less iterations as j increases, so its $\log(n)$. However, repeats n times, so $n\log(n)$.

$$f(n) = c_1 + (c_2 + c_3 + c_3 + c_4 + c_6 + c_7)n + (c_8 + c_9 + c_{10}) \cdot \log(n) \cdot n$$

$f(n)$ grows $n\log(n)$ as n grows large

Q5:

5.

$x = 0$

for $i \leftarrow 0$ to n **do** n

for $j \leftarrow 0$ to $(i \times n)$ **do** $n\frac{(n(n+1))}{2}$

$x = x + 10$ $n\frac{(n(n+1))}{2}$ ← $\sum_{i=1}^{n} i \cdot n = n\sum_{i=1}^{n} i = n\frac{(n(n+1)}{2}$

end for

end for

$$f(n) = c_1 + c_2 n + (c_3 + c_4)\, n\left(\frac{n(n+1)}{2}\right)$$

$f(n)$ grows n^3 as n grows very large