Aaron Stark
hw0913
https://github.com/aaronsta/COSC311/blob/master/hw0913
COSC 311, Fall 2018

R-4.2

A: 8n log n
B: 2n^2

Solve for n, n = 16.

For n0 = 17, B is always of greater value than A, so A is better than B for n0 = 17

R-4.3

A y = 40x^2
B y = 2x^3

Solve for x, x = 20

n0 = 21, after that B grows faster than A and thus A is better than B

R-4.8

Ordered by asymptotic growth rate

2^10
2^(log n)
3n + 100log(n)
4n
n*(log(n)
4n*log(n) + 2n
n^2 + 10n
n^3
2^n

R-4.9

example1

The loop runs n times (doing 1 primitive operation inside), added to a constant number of primitive operations outside the loop. So the method runs in

O(n)

R-4.10

example2

Constant number of primitive operations outside loop plus inside loop (which again has a constant number of primitive operations) runs floor(n/2) times, but 1/2 is a constant so again the number of overall operations is:
O(n)

R-4.11

example3
I worked out this table of the number of iterations of the inner loop for each value of j (i.e. outer loop value):

val of j	num of iterations of inner
0	1
1	2
2	3
...	...
n-1	n
n	n+1

So the primitive operation inside the inner loop runs
1+2+3+...+n + n+1 = n(n+1)/2
times, by Prop 4.3 of GtG book (Gauss) .
This if is of order

O(n^2)