Clayton Samson  
CSC 3102 – Homework 1

1.)

1. f13(*n*) = 210
2. f10(*n*) = log log *n*
3. f9(*n*) = log√(*n*)
4. f6(*n*)=log3*n*
5. f1(*n*)=*n*1/2
6. f2(*n*)=√(2*n*)
7. f3(*n*) = *n*+1
8. f7(*n*) = 7*n* log *n*
9. f11(*n*) = 6*n*4+9*n*3
10. f12(*n*) = 2log *n*
11. f8(*n*) = *n*log log *n*
12. f4(*n*) = 2n
13. f5(*n*) = 100n

2.) (n + 3)3 is *O*(n3) because (n + 3)3 is ≤ C \* n3 for C = 4 and N0 = 6, when using these values for C and N0

(n + 3)3 = 500 and 4 \* n3 = 864 and this holds as n increases.

3.) n2 is Ω(*n* log *n*) because n2 is ≥ *n* log *n* if C = 1 and N0 = 1

4.) if *d*(*n*) is O(*e*(*n*)) and *f*(*n*) is *O*(*g*(*n*)), then *d*(*n*)*f*(n) is O(*e*(*n*)*g*(*n*)) because if *d*(*n*) is ≤ *e*(*n*) for some C and N0 and *f*(*n*) is ≤ *g*(*n*) for some C and N0 then *d*(*n*)*f*(n) is ≤ *e*(*n*)*g*(*n*) for some C and N0.

5.) The code given is O(*n*2)

6.) for (j = m; j ≥ 0; j--)

A[j] = A[j-1];

A[0] = x;

7.) for (i = 0; i ≤ m; i++)

A[i] = A[i + 1];

8.) if(n < m-1)

For (j = n; j ≥ i; j--)

A[j] = A[j-1];

A[i] = x;

n++;

9.) for (i = 0; i < m; i++)

A[i] = A[i+1];

10.) int j = 0;

for (i = m; i ≥ 0; i--)

A[i] = B[j];

J++;