Problem 1

* 3n^3 – 7n^2 +100n = 36 is in O(n^3)
  + F(n) <= cG(n)
  + 3n^3 – 7n^2 +100n -36 <= 3n^3-7n^3+100n^3
  + <= 3n^3-7n^3+100n^3
  + <=96n^3
  + True, 3n^3 – 7n^2 +100n -36 is always less than 96n^3
* 10n+3log(n) is in O(n)
  + F(n) <= cG(n)
  + 10n + 3log(n) <= 10n+3n
  + <= 10n+3n
  + <=13n
  + True, 10n+3log(n) is always less than 13n
* n/1000 is in O(1)
  + F(n) <= cG(n)
  + n/1000 <= 1
  + As n approaches infinity, F(n) will approach infinity and will not remain constant
* log(n)^2 + log(n)/30 is in O(log(n)^2)
  + F(n) <= cG(n)
  + x^2 + x/30 <= 2x^2+x^2/30
  + <= 60x^2/30+x^2/30
  + <=61x^2/30
  + <=61log(n)^2/30 always bounds for any n greater than 1
* n^2/log(n)+3n is in O(n^2)
  + F(n) <= cG(n)
  + n^2/log(n)+3n <= 4n^2
  + F(n) will always be bound by no­ greater than 2 and c = 4

Problem 2

* 36n
  + O(n)
* (n^2)/2+15n
  + O(n^2)
* (n^2/4)(8/n)
  + O(n)
* n+10log(n)
  + O(n)
* 87262
  + O(1)
* n^4+2^n
  + 2^n

Problem 3

public int m1FindBiggest ( int [ ] a r r a y ) { //this method will print with O(n)

i f ( a r r a y . l e n g t h != 0 ) {

int v al u e = a r r a y [ 0 ] ;

for ( int i = 1 ; i < a r r a y . l e n g t h ; i ++){

i f ( a r r a y [ i ] > v al u e ) {

v al u e = a r r a y [ i ] ;

}

}

return v al u e ;

}

return −1;

}

// This one is a little tricky . //this method will print with O(n^2)

public void m 2P rin tT ri an gle ( int s i z e ) {

for ( int i = 1 ; i <= s i z e ; i ++){

for ( int j = 1 ; j <= i ; j ++){

System . out . p r i n t ( ‘ ‘ ∗ ’ ’ ) ;

}

System . out . p r i n t l n ( ) ;

}

}

// Assumption: each book has a t l e a s t 0 s t a r s and a t most 5 s t a r s

public void m3PrintBooks ( S t ri n g books [ ] , int [ ] s t a r s ) { //this method will print with O(n^2)

i f ( books . l e n g t h == s t a r s . l e n g t h ){

for ( int i = 0 ; i < books . l e n g t h ; i++) {

System . out . p r i n t ( books [ i ] + ‘ ‘ s s t a r s : ’ ’ ) ;

for ( int j = 0 ; j < s t a r s [ i ] ; j++) {

System . out . p r i n t ( ‘ ‘ ∗ ’ ’ ) ;

}

System . out . p r i n t l n ( ) ;

}

}

}