1. prove f(n) = log(n!) = O(nlogn) = Ω(nlogn)
2. Calculate time complexity

int fun(int n){  
 int sum = 0;   
 for(int i=1;i<=n;i\*=2)   
 sum += i\*i;   
 return i;   
}

int fun(int n)  
 if( n <= 1 )

return 1;   
 return fun( n/2 ) + n\*n;   
}

1. Analyze the recursion, find out the recurrence relation between cost functions of different input and solve the recurrence relation:

int fun(int n) {  
 int a,b;  
 if(n<=4)  
 return 0;  
 else {  
 a = n/2;  
 b = fun(n/4);  
 return a+b;  
 }

}

4.For each pair of f(n) and g(n) below, decide if f(n) = O(g(n)), f(n) = Ω(g(n)), or f(n) = Θ(g(n)). Justify your answer. Note that more than one of these relations may hold for a given pair; list all correct ones.

1. and.
2. g(n)=n!

5. find out the relationship between the pairs of functions. Determine whether *f* is O(g), Ω(g) or Ө(g).

