Problem 1 =

1) 10000000000n^2<=10000000001n^2| O(g(n))=O(n^2)
n^3<= 2n^3 | O(g(n))=O(n^3)
O(n^3) is asymptotically greater since when we are solving for big O seeing that n^3 is greater than 10000000000n^2
Another way is because the exponent is bigger

2) n^2*log(n) vs n*((log(n))^10)

n^2*log(n) is greater because of log law, since when multiply by log instead of n^2 we get log(n^2)+log(log(n)) vs log(n)+log(log(n)^10). 2 log (n)vs log(n) means that the left is greater than the right.

3) n^(log(n)) vs 2^(sqrt(2))
log(n)(log(n)) vs sqrt(n) log(2)
log((log(n))^2) vs log(sqrt(n) log(2))
2 (log(log(n))) vs log(sqrt(n))+log(log(2)) hence we state that 2^(sqrt(2) is greater.

4) These are equal since when we simplify both into big O We see that both functions are under the same O(2^n) hence when compared their constants which shows 2^2n is greater than 2^n

Problem 2 =

Best case = n is O(1)since if the case is even or multiple of 2 it only runs once and then it will exit the method

Worst case = O(sqrt(n)) because
- i*i=n
- i^2=n
- i=sqrt(n)

Average Case= O(sqrt(n)) because for defining theta we need for g(n) to be the same hence why theta takes the worst case which is the same.