1)

8n^2 = 64nlgn

n^2 = 8nlgn

n = 8lgn

n – 8lgn = 0

n = 43.56

insertion sort is faster than merge sort for values 0 < n < 43.56.

2)

	1 second	1 min	1 hour	1 day	1 month	1 year	1 century
Lg(n)	2^(1*10^6)	2^(60*10^6)	2^(3600*10^6)	2^(86400*10^6)	2^(2592000*10^6)	2^(31536000*10^6)	2^(3155760000*10^6)
Sqrt(n)	(11*10^6)^2	(60*10^6)^2	(3600*10^6)^2	(86400*10^6)^2	(2592000*10^6)^2	(31536000*10^6)^2	(3155760000*10^6)^2
N	1*10^6	60*10^6	3600*10^6	86400*10^6	2592000*10^6	31536000*10^6	3155760000*10^6
N*lg(n)	62746.1	2.8 * 10^6	1.33^(10^8)	2.75*(10^9)	7.18*(10^10)	7.97*(10^11)	6.86*(10^13)
N^2	(1*10^6)^(1/2)	(60*10^6)^(1/2)	(3600*10^6)^(1/2)	(86400*10^6)^(1/2)	(2592000*10^6)^(1/2)	(31536000*10^6)^(1/2)	(3155760000*10^6)^(1.2)
N^3	(1*10^6)^(1/3)	(60*10^6)^(1/3)	(3600*10^6)^(1/3)	(86400*10^6)^(1/3)	(2592000*10^6)^(1/3)	(31536000*10^6)^(1/3)	(3155760000*10^6)^(1/3)
2^n	Lg(1*10^6)	lg(60*10^6)	Lg(3600*10^6)	lg(86400*10^6)	lg(2592000*10^6)	lg(31536000*10^6)	lg(3155760000*10^6)
N!	9.45	11.16	12.79	13.9	15.25	16.15	17.757

3)

P(k) = $T(2^k) = 2^k \lg 2^k$

Base case: P(1) = $T(2^1) = (2^1)\lg(2^1) = 2$

Assume P(k) is true, $T(2^k) = 2^k \lg 2^k$

P(k + 1) = $2^{k+1} \lg 2^{k+1}$

Proof for k + 1:

$2T(2^{k+1} / 2) + 2^{k+1}$

$= (2)T(2^k) + 2^{k+1}$

$= (2)(2^k \lg 2^k) + 2^{k+1}$

$= 2^{k+1} \lg 2^k + 2^{k+1}$

$= 2^{k+1}(\lg 2^k + 1)$

$= 2^{k+1}(\lg 2^k + \lg 2)$

$= 2^{k+1} \lg 2^{k+1}$

4)

A: Omega

B: Omega

C: Theta

D: Omega

E: Theta

F: O

G: O

H: O

5)

```
Getminmax(X[])
Min = X[1]
Max = X[1]
For(i=2 to X length)
        If(X[i] > max)
                Max = X[1]
        If(X[i] < min)
```

Min = X[1]

Return(min,max)

Worst case this will have a runtime of 2N

6)

7)