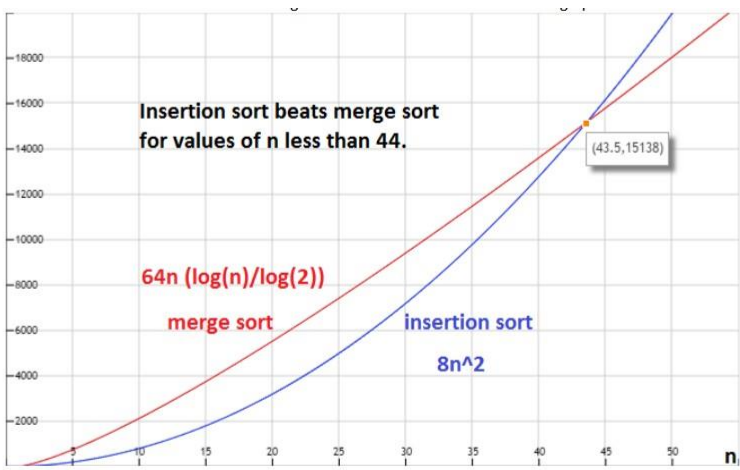
**CS 325 - HW 1 Solutions Examples**

**Problem 1: 1 point - must include either a graph, table or some explanation as to how they got the result 1 < n < 44 insertion sort runs faster than merge sort ( 2 ≤ n ≤ 43 )**




To solve this, I made a simple spreadsheet that quickly found that insertion sort above will beat for values of n < 44.

| value of n | 8n^2 | 64nlgn |
|---|---|---|
| 1 | 8 | 0 |
| 2 | 32 | 128 |
| 3 | 72 | 304 |
| 5 | 200 | 743 |
| 10 | 800 | 2126 |
| 20 | 3200 | 5532 |
| 40 | 12800 | 13624 |
| 43 | 14792 | 14933 |
| 44 | 15488 | 15374 |
| 45 | 16200 | 15817 |
| 50 | 20000 | 18060 |
| 100 | 80000 | 42521 |
| 1000 | 8000000 | 637810 |

2. **5 points - 0.5 point deduction for each one missed.**

1) (5 pts) For each of the following pairs of functions, either f(n) is O(g(n)), f(n) is Ω(g(n)), or f(n) = Θ(g(n)). Determine which relationship is correct and explain.

**CS 325 - HW 1 Solutions Examples**

a. **f(n) is O(g(n))** $f(n) = n^{0.25}$; $g(n) = n^{0.5}$

b. **f(n) is Ω(g(n))** $f(n) = n$; $g(n) = \log^2 n$

c. **f(n) is □(g(n))** $f(n) = \log n$; $g(n) = \ln n$

d. **f(n) is □(g(n))** $f(n) = 1000n^2$; $g(n) = 0.0002n^2 – 1000n$

e. **f(n) is O(g(n))** $f(n) = n\log n$; $g(n) = n\sqrt{n}$

f. **f(n) is O(g(n))** $f(n) = e^n$; $g(n) = 3^n$

g. **f(n) is □(g(n))** $f(n) = 2^n$; $g(n) = 2^{n+1}$

h. **f(n) is O(g(n))** $f(n) = 2^n$; $g(n) = 2^{2n}$

i. **f(n) is O(g(n))** $f(n) = 2^n$; $g(n) = n!$

j. **f(n) is O(g(n))** $f(n) = \lg n$; $g(n) = \sqrt{n}$

**CS 325 - HW 1 Solutions Examples**

3) a. **2 points: 1 for true (prove), 1 for proof**

If $f_1(n) = \Theta(g(n))$ and $f_2(n) = \Theta(g(n))$ then $f_1(n) = \Theta(f_2(n))$.

By definition if $f_1(n) = \Theta(g(n))$ then $c_1 g(n) \le f_1(n) \le c_2 g(n)$ for $n \ge n_0$ (I)

By definition if $f_2(n) = \Theta(g(n))$ then $k_1 g(n) \le f_2(n) \le k_2 g(n)$ for $n \ge n_2$ (II)

From (II) we obtain

$$g(n) \le (1/k_1) f_2(n) \Rightarrow c_2\, g(n) \le (c_2/k_1) f_1(n) \quad \text{(III)}$$

From (II) we can also obtain

$$(1/k_2) f_2(n) \le g(n) \Rightarrow (c_1/k_2)\, f_1(n) \le c_1 g(n) \quad \text{(IV)}$$

Now by combining inequalities (I) (III) and (IV) we obtain

$$(c_1/k_2)\, f_2(n) \le c_1 g(n) \le f_1(n) \le c_2 g(n) \le (c_2/k_1) f_2(n) \qquad \text{(V)}$$

If we let $c_3 = (c_1/k_2)$ and $c_4 = (c_2/k_1)$ then (V) becomes

$$c_3\, f_2(n) \le f_1(n) \le c_4 f_2(n) \quad \text{for } n_3 = \max\{n_0, n_2\}$$

Therefore by definition, $f_1(n) = \Theta(f_2(n))$.

b. **2 points: 1 for false (disprove), 1 for counterexample.**

If $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$ then $f_1(n) + f_2(n) = \Theta(g_1(n) + g_2(n))$

Many counterexamples:

If $f_1(n) = n$, $f_2(n) = n$, $g_1(n) = n!$ & $g_2(n) = n!$

**CS 325 - HW 1 Solutions Examples**

4) **10 points total**

README file – **1 point**
Fully commented code **- 1 points**

Run code on TEACH with the file - Execution
**4 points for the correct execution of insertion sort and 4 points for execution of merge sort.**

data.txt containing the values below
10 10 9 8 7 6 5 4 3 2 1
3 1 1 1
5 9 8 2 3 3

merge.out and insert.out each should contain
1 2 3 4 5 6 7 8 9 10
1 1 1
2 3 3 8 9

5) **10 points total- Solutions may vary**

a) **2 points** - Insertion Sort and Merge Sort code with timing added (you do not have to run)

b) **2 points** for data at **least 5 values for each algorithm that are non-zero**

Insertion Sort

| $x_2$ | $y_2$ |
|---|---|
| 0 | 0 |
| 10000 | 0.11 |
| 20000 | 0.45 |
| 30000 | 1.01 |
| 40000 | 1.82 |
| 50000 | 2.86 |
| 60000 | 4.1 |
| 70000 | 5.57 |
| 80000 | 7.24 |
| 90000 | 9.18 |

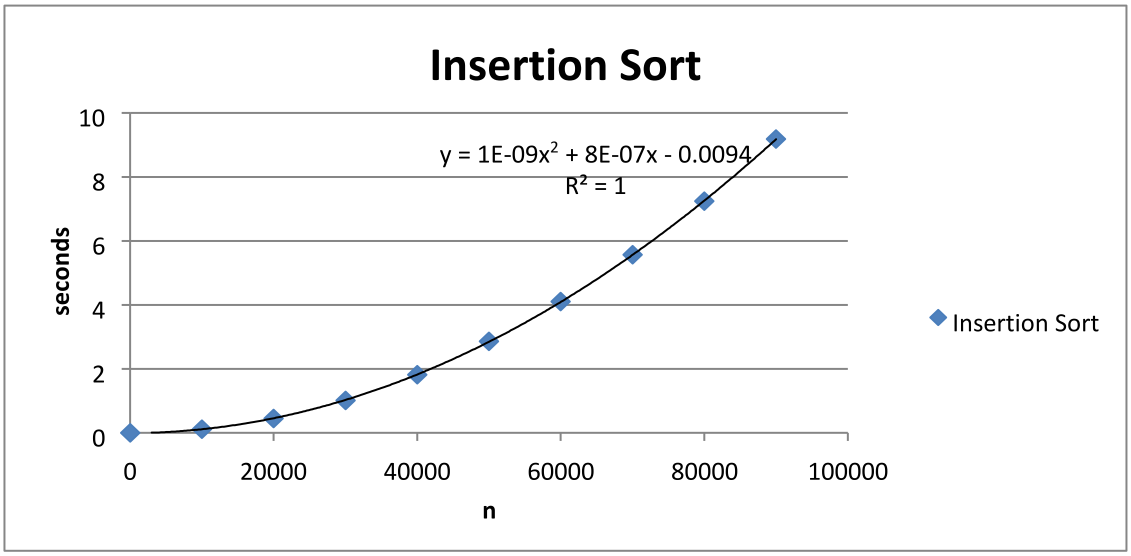
Merge Sort

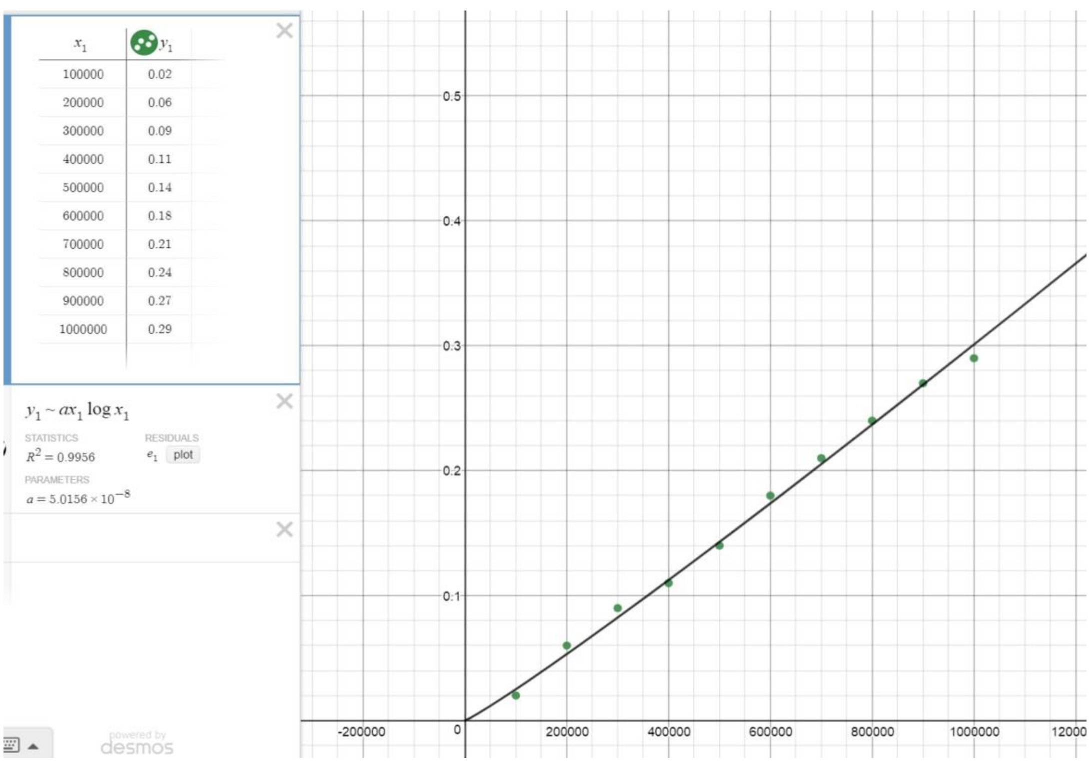
| n | Seconds |
|---|---|
| 0 | 0 |
| 100000 | 0.02 |
| 200000 | 0.06 |
| 300000 | 0.09 |
| 400000 | 0.11 |
| 500000 | 0.14 |
| 600000 | 0.18 |
| 700000 | 0.21 |
| 800000 | 0.24 |
| 900000 | 0.27 |
| 1000000 | 0.29 |

**CS 325 - HW 1 Solutions Examples**

c) **4 points** - Plot data and fit a curve Use Excel Matlab, R or Desmos

- Insertion sort plot 1 pt
- Insertion sort curve 1 pt
- Merge sort plot 1 pt
- Merge sort curve 1 pt



**CS 325 - HW 1 Solutions Examples**

Merge sort T(n) = (5.0156E-8)nlogn with $R^2$ = 0.9956

**Insertion sort** has a quadratic fitted curve is displayed on the graph,
T(n) = 1E-09$n^2$ + 8E-07n - 0.0094
$R^2$=1
This is an almost perfect fit. (Results may vary)

**Merge sort** looks linear (or nlogn) fitted line is on the graph. Full credit for fitting an nlgn curve

y = (5.0156E-8)nlogn
$R^2$ = 0.9956
This is also a very good fit.

d) **1 point for combined graph**

e) **1 point comparison**

**Insertion sort** – average experimental running time is $\Theta(n^2)$ which matches the theoretical value

**Merge sort** – average experimental running time is $\Theta(n)$ which differs from the theoretical value of $\Theta(n\lg n)$. Answers may vary.