CSE 250 Spring 2011
Homework 3
Due Date: March 21, Monday, by 2:05pm
Total Points: 30

Due Time: 2:05'00" pm.
Print your Name, UB #, and Recitation section on the first page.
If these guidelines are not followed, TA may deduct 10% of points.

1. (10 pts) Order the following functions by growth rate. Indicate which functions grow at the same rate. Use limit test to justify your answer.

$$N,\ \sqrt{N},\ N\log N,\ N^2/\log N,\ N\log(N^3)\ , 2^N, 2^{N/3}$$

Note: The easiest way to solve this problem is: First, arrange the functions in the order of increasing growth rate. Then, compare each pair of consecutive functions in the list.

Solution:

(1)

$$\sqrt{N},\ N,\ N\log N,\ N\log(N^3),\ N^2/\log N,\ 2^{N/3},\ 2^N$$

where $N\log N$ and $N\log(N^3)$ grow at the same rate.

(2) $\lim_{N\to\infty}\frac{\sqrt{N}}{N} = \lim_{N\to\infty}\frac{1}{\sqrt{N}} = 0$

$\lim_{N\to\infty}\frac{N}{N\log N} = \lim_{N\to\infty}\frac{1}{\log N} = 0$

$\lim_{N\to\infty}\frac{N\log N}{N\log(N^3)} = \lim_{N\to\infty}\frac{\log N}{3\log N} = \frac{1}{3}$

$\lim_{N\to\infty}\frac{N\log(N^3)}{N^2/\log N} = \lim_{N\to\infty}\frac{3\log N\log N}{N} = \lim_{N\to\infty}\frac{3(\log N)^2}{N} = \lim_{N\to\infty}\frac{6\log N\frac{1}{N}}{1} = \lim_{N\to\infty}\frac{6\log N}{N} = \lim_{N\to\infty}\frac{6}{N} = 0$

Because $N^2/\log N = o(N^3)$,

and $\lim_{N\to\infty}\frac{N^3}{2^{N/3}} = \lim_{N\to\infty}\frac{3N^2}{2^{N/3}\ln 2/3} = \lim_{N\to\infty}\frac{6N}{2^{N/3}\ln 2\ln 2/9} = \lim_{N\to\infty}\frac{6}{2^{N/3}\ln 2\ln 2\ln 2/27} = 0.$

So $N^2/\log N = o(2^{N/3})$

$\lim_{N\to\infty}\frac{2^{N/3}}{2^N} = \lim_{N\to\infty}\frac{1}{2^{2N/3}} = 0$

2. (3 pts.) An algorithm takes 2 ms (1 ms $= 10^{-3}$ seconds) for input size $n = 100$. How long will it take for the input size $n = 5000$ if the running time is the following (assume low-order terms are negligible)?

   (a) $\Theta(N)$,

   (b) $\Theta(N\log_2 N)$,

   (c) $\Theta(N^3)$

Solution:

(1) Suppose $f(N) = \Theta(N)$. This means that $f(N) = cN$ for some constant c. When $N = 100$, $f(N) = 2$ms. So

$$2 = c\cdot 100 \text{ which implies } c = 2\cdot 10^{-2} \text{ and } f(N) = 2\cdot 10^{-2}\cdot N$$

Plug in $N = 5000$, we have: $f(5000) = 2\cdot 10^{-2}\cdot 5000 = 100$ms.

(2) Suppose $g(N) = \Theta(N\log_2 N)$. This means that $g(N) = c\cdot N\log_2 N$ for some constant c. When $N = 100$, $g(N) = 2$ms. So

$2 = c \cdot 100 \cdot \log_2 100$, which implies $c = 2 \cdot 10^{-2} / \log_2 100 \approx 3.01 \cdot 10^{-3}$ and $g(N) = 3.01 \cdot 10^{-3} \cdot N \cdot \log_2 N$. Plug in $N = 5000$, we have: $g(5000) = 3.01 \cdot 10^{-3} \cdot 5000 \cdot \log_2 5000 = 185$ms.

(3) Suppose $h(N) = \Theta(N^3)$. This means that $h(N) = c \cdot N^3$ for some constant c. When $N = 100$, $h(N) = 2$ms. So

$$2 = c \cdot 100^3 \text{ which implies } c = 2 \cdot 10^{-6} \text{ and } h(N) = 2 \cdot 10^{-6} \cdot N^3$$

Plug in $N = 5000$, we have: $h(5000) = 2 \cdot 10^{-6} \cdot (5000)^3 = 250000ms = 250sec = 4.17$min.

3. (3 pts.) An algorithm takes 2 ms for input size $n = 100$. How large a problem can be solved in 10 min if the running time is the following (assume low-order terms are negligible)?

   (a) $\Theta(N)$,

   (b) $\Theta(N \log_2 N)$,

   (c) $\Theta(N^3)$

   Solution:

   Note that 10 mins = 600 sec = $6 \cdot 10^5$ms.

   (1) From problem 2 (1), we already know: $f(N) = 2 \cdot 10^{-2} \cdot N$ms. We need to solve:

   $$6 \cdot 10^5 = 2 \cdot 10^{-2} \cdot N \text{ for } N$$

   So $N = 3 \cdot 10^7$.

   (2) From problem 2 (2), we already know: $g(N) = 3.01 \cdot 10^{-3} N \cdot \log_2 N$ms. We need to solve:

   $$6 \cdot 10^5 = 3.01 \cdot 10^{-3} N \log_2 N \text{ for } N$$

   There is no formula for solving such an equation. You can use a calculator to try to find approximate solution: $N \approx 8.5 \cdot 10^6$.

   (3) From problem 2 (3), we already know: $h(N) = 2 \cdot 10^{-6} \cdot N^3$. We need to solve:

   $$6 \cdot 10^5 = 2 \cdot 10^{-6} \cdot N^3$$

   for N. So $N^3 = 3 \cdot 10^{11}$ and $N \approx 6694$.

4. (1+1=2 pts) Consider the following program segments:

   (a)

```
int s = 0;
for (int i = 0; i < n ; i++)
    for (int j = 0; j < n ; j++)
         for  (int k = 0; k < n ; k++)
            if ((i+j) == k))
                 s = s+i;
```

How many times the inner loop body iterates? What's the growth rate of the run time function of this segment.

Solution: The number of loop iterations is n^3. Since the loop body takes constant number of basic instructions, the runtime of the loop is $\Theta(n^3)$ and its growth rate is n^3.

(b)

```
int t = 0;
for (int i = 0; i < n ; i++)
    for (int j = n-1; j >= i; j--)
            s = s+i+j;
```

How many times the inner loop body iterates? What's the growth rate of the run time function of this segment.

Solution:

The number of loop iterations is: $n + (n-1) + \ldots + 1 = n(n-1)/2$.

Since the loop body takes constant number of basic instructions, the runtime of the loop is $\Theta(n^2)$ and its growth rate is n^2.

5. (3+3+3+3= 12 pts) Consider the array:

1, -9, 89, 25, 0, 5, 99, -4

Show the progress and the array after each pass, when the following sorting algorithms are performed on this array.

(a) Selection Sort
(b) Bubble Sort
(c) Insertion Sort
(d) Merge Sort

Solution:

(a) by Selection Sort

1, -9, 89, 25, 0, 5, 99, -4
-9, 1, 89, 25, 0, 5, 99, -4
-9, -4, 89, 25, 0, 5, 99, 1
-9, -4, 0, 25, 89, 5, 99, 1
-9, -4, 0, 1, 89, 5, 99, 25
-9, -4, 0, 1, 5, 89, 99, 25
-9, -4, 0, 1, 5, 25, 99, 89
-9, -4, 0, 1, 5, 25, 89, 99

(b) by Bubble Sort

1, -9, 89, 25, 0, 5, 99, -4
-9, 1, 25, 0, 5, 89, -4, 99
-9, 1, 0, 5, 25, -4, 89, 99
-9, 0, 1, 5, -4, 25, 89, 99
-9, 0, 1, -4, 5, 25, 89, 99
-9, 0 ,-4, 1, 5, 25, 89, 99
-9, -4, 0, 1, 5, 25, 89, 99

(c) by Insertion Sort

1, -9, 89, 25, 0, 5, 99, -4
-9, 1, 89, 25, 0, 5, 99, -4
-9, 1, 89, 25, 0, 5, 99, -4
-9, 1, 25, 89, 0, 5, 99, -4
-9, 0, 1, 25, 89, 5, 99, -4
-9, 0, 1, 5, 25, 89, 99, -4
-9, 0, 1, 5, 25, 89, 99, -4
-9, -4, 0, 1, 5, 25, 89, 99

(d) by Merge Sort

1, -9, 89, 25, 0, 5, 99, -4
-9, 1, 25, 89, 0, 5, -4, 99
-9, 1, 25, 89, -4, 0, 5, 99
-9, -4, 0, 1, 5, 25, 89, 99