CSE/EE 373 Autumn 1994
Data Structures and Algorithms
Assignment # 2 Common Mistake Solutions

1. Problem 2.1 on page 36.

   Answer:

   Order from small to big:

   $$\frac{2}{n}, 37, n^{\frac{1}{2}}, n, n\log\log n, n\log n, n\log n^2, n(\log n)^2, n^{\frac{3}{2}}, n^2, n^2\log n, n^3, 2^{\frac{n}{2}}, 2^n$$

f	f'
$\frac{2}{n}$	$\frac{-2}{n^2}$
37	0
$n^{\frac{1}{2}}$	$\frac{1}{2}n^{\frac{-1}{2}}$
n	1
$n\log\log n$	$\log\log n + (\log e)^2\frac{1}{\log n}$
$n\log n$	$\log n + \log e$
$n\log n^2$	$2\log n + 2\log e$
$n(\log n)^2$	$\log n(\log n + 2\log e)$
$n^{\frac{3}{2}}$	$\frac{3}{2}n^{\frac{1}{2}}$
n^2	$2n$
$n^2\log n$	$n(2\log n + \log e)$
n^3	$3n^2$
$2^{\frac{n}{2}}$	$\frac{1}{2}2^{\frac{n}{2}}\ln 2$
2^n	$2^n\ln 2$

2. Problem 2.6(c) on page 36.

   Answer:

   (1) $O(n)$;
   (2) $O(n^2)$;
   (3) $O(n^3)$;
   (4) $O(n^2)$;
   (5) $O(n^5)$;
   (6) $O(n^4)$;

   Explanation:

When i is 2, j could only be 2 to go into the $3rd$ for loop; when i is 3, j could be 3 or 6 to execute the $3rd$ for loop, and when i is 4, j could only be 4, 8, and 12 to satisfied the condition of the $3rd$ for loop; etc. Therefore, for any i, the biggest number that j could ever reach to be a multiple of i and also be less than i^2 is $i * (i - 1)$.
Following are some concrete examples:

i	j
2	2
3	3, 6
4	4, 8, 12
5	5, 10, 15, 20
6	6, 12, 18, 24, 30
...	

Notice, when $i = 1$, $sum++$ is not executed, because $j < i^2$ is false.
Total time of the third for loop - $sum++$ gets to executed is:

$$T \simeq n(n^2 + n(n^2)) \simeq n^3 + n^4 = O(n^4)$$