CS182 Spring 2016: Homework 4

Due date: Friday, March 4, 2016 (before class).

1. (4 pts)Let $f(n) = 3n^2 + 5n + 2$ be the running time of Steve's wonderful sorting algorithm. Show that f(n) is $O(n^2)$ by identifying constants C and k such that $f(n) \leq Cn^2$ whenever $n > k$. Prove that your chosen C and k work correctly.

   Soln: C = 10; k = 1. Proof: for all $n > 1$, $f(n) = 3n^2+5n+2 < 3n^2+5n^2+2n^2 = 10n^2$.

2. (4 pts)Let $f(n) = 100n^2 logn + 2n^3 - 1000$. Let g(n) be the most slowly growing function of n consisting of a single term with a coefficient of 1 such that f(n) is O(g(n)). Identify g(n) and prove that f(n) is O(g(n)) by giving a C and a k and showing that they yield the desired result.

   Soln: f is $O(n^3)$. Proof: Let k = 1. For $n > 1$, $f(n) = 100n^2 logn + 2n^3 - 1000 < 100n^2 n + 2n^3 = 102n^3$. Thus we can let C = 102 and it will be true that for all $n > k$, $f(n) \leq Cn^3$.

3. (8 pts)Consider the following functions:

   (a) $f(n) = 2^n$
   (b) $g(n) = n!$
   (c) $h(n) = n^{logn}$

   Which of the following statements about the asymptotic behavior of f(n), g(n), and h(n) are true?

   (a) $f(n) = O(g(n))$;
   (b) $g(n) = O(h(n))$;
   (c) $f(n) = \Omega(g(n))$;
   (d) $g(n) = O(f(n))$;
   (e) $h(n) = O(f(n))$;
   (f) $g(n) = \Omega(f(n))$;

   Soln: (a), (e) and (f). According to order of growth: $h(n) < f(n) < g(n)$
   (g(n) is asymptotically greater than f(n) and f(n) is asymptotically greater than h(n)).
   We can easily see above order by taking logs of the given 3 functions.
   $lognlogn < n < log(n!)$ (logs of the given f(n), g(n) and h(n)).
   Note that $log(n!) = \theta(nlogn)$
   Thus correct solutions are (a), (e) and (f).

4. (8 pts)What is the time complexity of fun() for the following cases?

(a)
```
int fun(int n)
{
  int count = 0;
  for (int i = 0; i < n; i++)
     for (int j = i; j > 0; j--)
        count = count + 1;
  return count;
}
```

(b)
```
void fun(int n, int arr[])
{
    int i = 0, j = 0;
    for(; i < n; ++i)
        while(j < n && arr[i] < arr[j])
            j++;
}
```

Soln:
(a) O(n^2). The time complexity can be calculated by counting number of times the expression "count = count + 1;" is executed. The expression is executed 0 + 1 + 2 + 3 + 4 + ... + (n-1) times. Time complexity = $\Theta(0+1+2+3+..+n-1) = \Theta(n*(n-1)/2) = \Theta(n^2)$

(b) O(n). In the first look, the time complexity seems to be $O(n^2)$ due to two loops. But, please note that the variable **j** is not initialized for each value of variable i. So, the inner loop runs at most n times.

5. (4 pts)In a competition, four different functions are observed. All the functions use a single for loop and within the for loop, the same set of statements are used. Consider the following for loops:

   (a) $for(i = 0; i < n; i++)$
   (b) $for(i = 0; i < n; i+ = 2)$
   (c) $for(i = 1; i < n; i* = 2)$
   (d) $for(i = n; i > -1; i/ = 2)$

   If n is the size of input(positive), which function is most efficient(if the task to be performed is not an issue). Explain your response.

   Soln: (C) The time complexity of first for loop is O(n).
   The time complexity of second for loop is O(n/2), equivalent to O(n) in asymptotic analysis.

The time complexity of third for loop is O(logn).
The fourth for loop doesn't terminate.
Thus, the correct response is (C).

6. (6 pts)

   (a) Express the binary number 101101 in decimal and hexadecimal.

   (b) Express the hexadecimal number 9D3C in binary and decimal.

   (c) Express the decimal number 7256 in binary and hexadecimal.

   Soln:
   $a.(101101)_2 = (45)_{10} = (2D)_{16}$
   $b.(9D3C)_{16} = (1001110100111100)_2 = (40252)_{10}$
   $c.(7256)_{10} = (1110001011000)_2 = (1C58)_{16}$

7. (4 pts)George has 44 boxes of soda in his truck. The cans of soda in each box are packed oddly so that there are 113 cans of soda in each box. George plans to pack the sodas into cases of 12 cans to sell. After making as many complete cases as possible, how many sodas will George have leftover?

   Soln: First, we note that:
   $44 \equiv 8 \pmod{12}$
   $113 \equiv 5 \pmod{12}$
   Thus, $44 \cdot 113 \equiv 8 \cdot 5 \equiv 40 \equiv 4 \pmod{12}$, meaning there are 4 sodas leftover.

8. (8 pts)What are the tens and units digits of 7^{1942}?

   Soln: Since we want only the tens and units digits of the number in question, it suffices to find the remainder when the number is divided by 100. In other words, all of the information we need can be found using arithmetic mod 100.

   We begin by writing down the first few powers of 7 mod 100:

   $7, 49, 43, 1, 7, 49, 43, 1, \ldots$

   A pattern emerges! We see that $7^4 = 2401 \equiv 1$ (mod 100). So for any positive integer k, we have $7^{4k} = (7^4)^k \equiv 1^k \equiv 1$ (mod 100). In particular, we can write

   $7^{1940} = 7^{4 \cdot 485} \equiv 1$ (mod 100).

   By the "multiplication" property, it follows that

   $7^{1942} = 7^{1940} \cdot 7^2 \equiv 1 \cdot 7^2 \equiv 49$ (mod 100).

   Therefore, by the definition of congruence, 7^{1942} differs from 49 by a multiple of 100. Since both integers are positive, this means that they share the same tens and units digits. Those digits are 4 and 9, respectively.

9. (4 pts)Is it possible to find a number that is both a multiple of 2 but not a multiple of 4 and a perfect square? Please explain.

   Soln: No, you cannot. Rewriting the question, we see that it asks us to find an integer n that satisfies $4n + 2 = x^2$.

   Taking mod 4 on both sides, we find that $x^2 \equiv 2 \pmod 4$. Now, all we are missing is proof that no matter what x is, x^2 will never be a multiple of 4 plus 2, so we work with cases:

   $x \equiv 0 \pmod 4 \implies x^2 \equiv 0 \pmod 4$

   $x \equiv 1 \pmod 4 \implies x^2 \equiv 1 \pmod 4$

   $x \equiv 2 \pmod 4 \implies x^2 \equiv 4 \equiv 0 \pmod 4$

   $x \equiv 3 \pmod 4 \implies x^2 \equiv 9 \equiv 1 \pmod 4$

   This assures us that it is impossible to find such a number.