Student ID: 1512002
Name: Le Duong Tuan Anh

Chapter 1 – Homework – Week 01

Problem 1:

Let f(n) = $\frac{n^5}{5} - 10n^4 - \frac{n}{2} + 300$.   f(n) = O($n^5$).

Select that c = 5; n_0 = 2; g(n) = n^5

=> f(n) ≤ c.g(n) ∀n ≥ n_0 (*)

Prove (*):

We have: c.g(n) – f(n) = $\boldsymbol{\frac{24n^5}{5} + 10n^4 + \frac{n}{2} - 300 \geq \frac{73}{5}\ \forall n \geq 2}$

So, f(n) = O(n^5)

Problem 2:

The statement "The running time of algorithm A is at least O(n^2)" is meaningless, because of phrase "at least".

The algorithm A with time complexity O(n^5) is more complex than algorithm B with O(n^2), but both will have running time at least O(n^2). For example, f(n) = $\frac{n^5}{5} - 10n^4 - \frac{n}{2} + 300$ in problem 1 above, we proved that f(n) = O(n^5). However, we say that "algorithm running time with f(n) is at least O(n^2)" is not wrong. But it's meaningless because O(n^2) is very different with O(n^5).

Problem 3:

a. f(x) = $2x^2 - x + 30$ is O(x^2)

   Select: c = 10; x_0 = 3; g(x) = x^2

   => f(x) ≤ c.g(x) ∀x ≥ x_0 (*)

   Prove (*):

   We have: c.g(x) – f(x) = $8x^2 + x - 30 \geq 45\ \forall x \geq 3$

   So, f(x) = O(x^2)

b. f(x) = $(3x + 2)\log_2(x^2 + 5)$ is O($x\log_2 x$)

   Select: c = 13; x_0 = 1; g(x) = $x\log_2 x$

   => f(x) ≤ c.g(x) ∀x ≥ x_0 (*)

Prove (*):

We have:

c.g(x) – f(x) $= 13x\log_2 x - 3x\log_2(x^2+5) - 2\log_2(x^2+5)$

$= 13x\log_2 x - 3x\log_2(x^2+5) - 2\log_2(x^2+5)$

$= 3x(3\log_2 x - \log_2(x^2+5)) + 2(2x\log_2 x - \log_2(x^2+5))$

$= 3x.\log_2\frac{x^3}{x^2+5} + 2\log_2\frac{x^{2x}}{x^2+5} \geq 0\ \forall x \geq 1$

So, $f(x) = O(x\log_2 x)$

c. $f(x) = (x^2 + 4\log_2 x)/(x+1)$ is $O(x)$

Select: $c = 4$; $x_0 = 1$; $g(x) = x$

=> $f(x) \leq c.g(x)\ \forall x \geq x_0$ (*)

Prove (*)

We have:

$c.g(x) - f(x) = 4x - \frac{x^2+4\log_2 x}{x+1} = \frac{3x^2+4x-4\log_2 x}{x+1} = \frac{3x^2+4.\log_2\frac{2^x}{x}}{x+1} \geq 0\ \forall x \geq 1$

So, $f(x) = O(x)$

Problem 4:

a. $f(x) = 10$ is not $O(x)$

Select: $c = 10$; $x_0 = 0$; $g(x) = 1$

=> $f(x) \leq c.g(x)\ \forall x \geq x_0$

$(10 \leq 10*1\ \forall x \geq 0)$

=> $f(x)$ is $O(1)$

O(1) is independent with x (O(1) is constant), so f(x) is not O(x)

b. $f(x) = 3x + 7$ is $O(x)$

With $c = 4$; $x_0 = 10$, $g(x) = x$

We have: $f(x) \leq c.g(x)\ \forall x \geq x_0$ (*)

Prove (*):

$f(x) - c.g(x) = x - 7 \geq 0\ \forall x \geq 10$

So, f(x) is O(x)

c. $f(x) = 2x^2 + 2$ is not $O(x)$

Assume that f(x) is O(x), there will exist c and x_0 that, with $g(x) = x$, we have:

$f(x) \leq c.g(x) \quad \forall x \geq x_0.$

$\Leftrightarrow 2x^2 + 2 \leq c.x \quad \forall x \geq x_0.$

$\Leftrightarrow 2x + 2/x \leq c \quad \forall x \geq x_0.$

Because c is constant, x is variable so $\forall c \in R, \exists x \text{ that } 2x + \frac{2}{x} \geq c$

=> f(x) is not O(x)

Problem 5:

a. $3^{n+3} = O(3^n)$ is TRUE.

   We have: $3^{n+3} = 9.3^n \leq 10.3^n\ \forall n \geq 0$

   With c = 10, $n_0 = 0$; $g(n) = 3^n$

   $3^{n+3} \leq c.g(n)\ \forall n \geq n_0$.

   <=> $3^{n+3} = O(3^n)$

b. $3^{3n} = O(3^n)$ is FALSE.

   Let $f(n) = 3^{3n}$. Assume that $f(n) = O(3^n)$, there will exist c and x_0 that, with $g(n) = 3^n$ we have: $f(n) \leq c.g(n)\ \forall n \geq n_0$.

   <=> $3^{3n} \leq c.3^n$

   <=> $(3^3)^n \leq c.3^n$

   <=> $9^n \leq c$

   <=> $n.\log 9 \leq \log(c)/\log(9)$

   <=> $n \leq \log(c) / \log^2(9)$

   So for every $n \geq \log(c)/\log^2(9)$, $3^{3n} \geq c.3^n$.

   Therefore, 3^{3n} is not $O(3^n)$.

Problem 6:

a. Function count the number of an element x in an array of n integers.

   (assume array **a[]** with **n** integers).

```
int count_num(int a[], int n, int x){
    if (n == -1) return 0;
    if (a[n] == x) return count_num(a, n-1, x) + 1;
    return count_num(a, n-1, x);
}
```

b. Function find the n^{th} Fibonacci number.

   (start from 1)

```
int fibo(int n){
    if (n == 1 || n == 2) return 1;
    return fibo(n-1) + fibo(n-2);
}
```

Problem 7:

In real life applications, correctness is more important than performance, because:

- The achivement of solving problem is assessed based on result. If the answer is wrong, it will be meaningless.
- We can wait for a long time to solve problem correctly, but we are not allowed to be fast and then make a wrong result. It will affect the entire process.
- For example, for a Banking Sector, if the transaction fails because of wrong program, this will lead to many problems then.

Problem 8:

We need to study algorithms and perfomance for those things:

- We will know many different algorithm, about searching, sorting, storing data, etc. in order to apply it in coding and fasten our applications.
- We will able to analyze and compare many algorithms to find out which algo. is the best for use.
- We can not only improve the current algorithms but also make other algorithms that can work better.

Problem 9:

Problem: Print the reversed version of number X.

Example:

Input: 123 *Output: 321*

Code (C++):

```
void write_reverse(int x){
	if ( x < 10) cout << x;
	else{
		write_reverse( x % 10 );
		cout << x/10;
	}
}
```

Problem 10:

Code (C++):

```
void write_reverse_iteration(int x){
        int ans = 0;
        while (x > 0){
                ans = ans * 10 + x % 10;
                x / = 10;
        }
        cout << ans; //Print answer
}
```