Homework 1

CSE 321 – Introduction to Algorithms – Fall 2021

1801042103 – Ozan GECKIN

① Use the formal definitions of asymptotic notations to determine whether the following statements are true or not.

ⓐ $(2^n + n^3) \in O(4^n) \Rightarrow$

Rule → O-notation

$O(g(n)) = \{f(n)\}$ → there is exist positive const c and n_0 such that;

$0 \le f(n) \le c.g(n)$ all $n > n_0$

$O(4^n) = \{2^n + n^3 ; 0 \le (2^n + n^3) \le c.4^n$

any positive to the constant c, the equation is still valid

Examples

$0 \le (2^n + n^3) \le 1.4^n$ → We can compare 2^n with 4^n because of $2^n > 3^n$

$4^n = 2^{2n}$ is greather than 2^n

Thus, a is True.

ⓑ $\sqrt{10n^2 + 7n + 3} \in \Omega(n) \Rightarrow$

Rule → Ω-notation

$\Omega(g(n)) = \{f(n)$: there is exist positife const c and n_0 such that;

$0 \le c.g(n) \le f(n)$ for all $n \ge n_0$

$c.n \le \sqrt{10n^2 + 7n + 3}$ $c > 0$, $n \ge n_0$

For omega(Ω) notation, $\sqrt{10n^2+7n+3}$ grows at the same rate or faster than n.

$\underbrace{\sqrt{10n^2+7n+3} = \sqrt{n^2} = n}$

→ Eliminate constant factor
→ Eliminate low-order factor

$c = 1$, $n_0 \ge 1$ is prove that this expression is true.

Thus, a is True.

1801042103 - Ozan GEÇKİN

ⓒ $n^2+n \in o(n^2)$

Rule → o-notation

$o(g(n)) = f(n)$: for all constants $c>0$, there exists a constant $n_0>0$; such that $0 \leq f(n) \leq c.g(n)$ for all $n \geq n_0$

$n^2+n < c.n^2$ $\quad c>0, n \geq n_0$

→ n^2 must grows greather than n^2+n. But that's not true.

Eliminate low-order term $n^2+n \to n^2$

(n^2+n) grow rate of n^2

Thus, the expression is False. (can be like $n^2+n \in \Theta(n^2)$)

ⓓ $3\log_2^2 n \in \Theta(\log_2 n^2)$

Rule → Θ notation

$\Theta(g(n)) = \{f(n)$: there exists positive constants c_1, c_2 and n_0 such that; $0 \leq c_1.g(n) \leq f(n) \leq c_2.g(n)$ for all $n > n_0\}$

→ $c_1.\log_2 n^2 \leq 3\log_2^2 n \leq c_2.\log_2 n^2$ $\quad c_1, c_2 > 0, n \geq n_0$

This inequality can not be achieved.

→ This expression, "$3\log_2^2 n$" grows at the same rate "$\log_2 n^2$". But that is not true.

→ $\log_2^2 n = \log_2(\log_2 n) \neq \log_2 n . \log_2 n$

→ $3\log_2^2 n$ ⇒ "Eliminate constant factor"

→ $\log_2^2 n \to \log_2(\log_2 n)$

→ grow rate of $\log_2 n^2 \to \log_2 n$

Thus; $3\log_2^2 n \notin \Theta(\log_2 n^2)$, so $3\log_2^2 n \in o(\log_2 n^2)$ is True

this expression $\boxed{\text{False}}$

180104103-Oğuz GÖKIN

ⓔ $(n^3+1)^6 \in O(n^3)$

Rule → Big-O notation

$O(g(n)) = f(n)$ there exists positive constants c and n_0 such that;

$0 \leq f(n) \leq c_1 g(n)$ for all $n \geq n_0$

→ $(n^3+1)^6 \leq c.n^3$

$n^{18} + x_1 n^{17} + x_2 n^{16} + \dots + 1 \leq c.n^3$

$n^{18} < n^3$ it is false

Thus $(n^3+1)^6 \in O(n^3)$ statement is false

Question 2

② Use the formal definition of Θ notation to find $\Theta(g(n))$ class the following functions belong to. Give the simplest $g(n)$ possible in your answers.

a) $2n \log(n+2)^2 + (n+2)^2 \log \frac{n}{2} =$

$$= \underbrace{4n \log(n+2)}_{\in \Theta(n\log n)} + \underbrace{(n+2)^2 (\log n - \log 2)}_{\in \Theta(n^2 \log n)}$$

grow rate of $(n^2 \log n)$

Thus, $\Theta(n\log n) + \Theta(n^2 \log n) \in \Theta(n^2 \log n)$

$\Theta(g(n)) = \Theta(n^2 \log n) \Rightarrow g(n) = n^2 \log n$

ⓑ $\underbrace{0,001 n^4}_{\in \Theta(n^4)} + \underbrace{3n^3}_{\in \Theta(n^3)} + \underbrace{1}_{\in \Theta(1)} \Rightarrow \Theta(n^4) + \Theta(n^3) + \Theta(1) \in \Theta(n^4)$

Thus, $\Theta(g(n)) = \Theta(n^4) \Rightarrow g(n) = n^4$

Question 3 1801042103 - Ozan GEÇKİN

(3) Compare and sort the following functions in terms of their orders of growth by using limit approach.

a) $\log n$, $n^{\log n}$, $n^{1.5}$

$$\rightarrow \lim_{n\to\infty} \frac{\log n}{n^{\log n}} \Rightarrow \lim_{n\to\infty} \frac{\frac{1}{n\ln 2}}{\frac{2n^{\log n-1}\ln n}{\ln 2}} = \frac{\frac{1}{n}}{2n^{\log n-1}\ln n} = \frac{1}{2n^{\log_2 n}\ln n} = \frac{1}{\infty} = 0$$

rule = result $0 \Rightarrow$ faster at the top

result $\infty \Rightarrow$ faster at the bottom

$$\rightarrow \lim_{n\to\infty} \frac{n^{\log n}}{n^{1.5}} = \frac{1}{n^{1.5}} \cdot \lim_{n\to\infty} (n^{\log n}) = \frac{1}{n^{1.5}} \infty = \infty$$

Thus, $n^{\log n} > n^{1.5} > \log n$

b) $n!$, 2^n, n^2 (Use Stirling's formula for $n!$)

rule = Stirling's Formula

$$n! \cong \sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n$$

$$\rightarrow \lim_{n\to\infty} \frac{n!}{2^n} = \lim_{n\to\infty} \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{2^n} = \lim_{n\to\infty} \sqrt{2\pi n}\left(\frac{n}{2e}\right)^n = \infty \quad n! > 2^n$$

$$\rightarrow \lim_{n\to\infty} \frac{2^n}{n^2} = \lim_{n\to\infty} \frac{\ln 2 . 2^n}{2n} = \lim_{n\to\infty} \frac{(\ln 2)^2 . 2^n}{2} = \infty \quad 2^n > n^2$$

Thus, $n! > 2^n > n^2$

c) $n\log n$, $\sqrt{n}$

$$\rightarrow \lim_{n\to\infty} \frac{n\log n}{\sqrt{n}} = \lim_{n\to\infty} n^{1-1/2} . \log n = \infty$$

Thus, $n\log n > \sqrt{n}$

d) $n2^n, 3^n$

$$\lim_{n\to\infty}\frac{3^n}{n2^n} = \lim_{n\to\infty}\frac{\left(\frac{3}{2}\right)^n}{n} = \lim_{n\to\infty}\left(\frac{3}{2}\right)^n \cdot \ln\frac{3}{2}$$

$$= \ln 3 - \ln 2) \cdot \lim_{n\to\infty}\left(\frac{3}{2}\right)^n = \infty$$

Thus, $3^n > n.2^n$

e) $\sqrt{n+10}, n^3$

$$\lim_{n\to\infty}\frac{n^3}{\sqrt{n+10}} = \lim_{n\to\infty}\frac{n^3}{\sqrt{n}\cdot\sqrt{1+\frac{10}{n}}} = \lim_{n\to\infty}\frac{n^{5/2}}{\sqrt{1+\frac{10}{n}}} = \frac{\lim_{n\to\infty} n^{5/2}}{\lim_{n\to\infty}\sqrt{1+\frac{10}{n}}} = \infty \cdot \frac{1}{\lim_{n\to\infty}\sqrt{1+\frac{10}{n}}}$$

$$= \infty \cdot \frac{1}{\sqrt{\lim_{n\to\infty} 1 + \lim_{n\to\infty}\frac{10}{n}}} = \infty \frac{1}{\sqrt{1+\left(10 . \lim_{n\to\infty}\frac{1}{n}\right)}} = \infty . \frac{1}{\sqrt{1+10 \frac{\lim_{n\to\infty} 1}{\lim_{n\to\infty} n}}}$$

$$= \infty \frac{1}{\sqrt{1+\frac{10}{\underbrace{\lim_{n\to\infty} n}_{\infty}}}} = \infty \frac{1}{1} = \infty$$

Thus, $n^3 > \sqrt{n+10}$

Question 4 180104 2103 - Ozan GEGKIN

④ Consider the worst case of the following algorithm.

a) Basic operation is $B[i,j] != B[j,i]$ the <u>comparison</u>

b) $$\sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} [(n-1)-(i-1)+1]$$

$$= \sum_{i=0}^{n-2} (n-1+i) = (n-1)+(n-2)+ \cdots +1 = \frac{(n-1).n}{2}$$

The algorithm most executed $\frac{n(n-1)}{2}$ times

c) $$\sum_{i=0}^{n-2} (n-1-i) = \frac{(n-1)n}{2} = \frac{n^2-n}{2}$$

Time complexity is $O(n^2)$

Question 5

⑤ Consider the following algorithm.

a) Basic operations is $C[i,j] = C[i,j] + A[i,k] * B[k,j]$

b) $$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} 1 = (n-1)n^2 \text{ times } \quad \underline{n^3}$$

c) $(n-1)n^2 = n^3 - n^2$

Time complexity is $O(n^3)$

1801042103 - Ozan GEÇKİN

Question 6

6) Design an algorithm that finds and prints all pairs whose multiplication yields the desired number in an unordered array. for example, let the array be {1,2,3,6,5,4} and let the desired number be 6. Then, our pairs will be (1,6) and (2,3). Write your algorithm as pseudo-code, and find the time complexity of the algorithm.

```
findPairs (array[], arraySize, desiredNumber)

    for i=0 to arraySize do:
     for j=i to arraySize do:
        if (array[i] * array[j] == desiredNumber)
            print (array[i], array[j])
```

Time Complexity =

$$\sum_{i=0}^{n} \sum_{j=1}^{n} 1 = \frac{n.n}{2} = \frac{n^2}{2} \Rightarrow O(n^2)$$