31/10/2021

CSE 321 - Homework 1.

Yusuf Arslan

Q-1

We know that:

$O(1) \subset O(\log(\log n)) \subset O(\log n) \subset O(n^a) \subset O(n) \subset O(n \log n) \subset O(n^a) \subset O(a^n)$ (a > 1) $\subset O(n a^n) \subset O(n!) \subset O(n^{n^b})$ (b > 1)

Let simplify the functions

- $T_1 \in O(\log n)$
- $T_2 \in O(\log(\log n))$
- $T_3 \in O(n^5)$
- $T_4 \in O(n)$
- $T_5 \in O(n^2)$
- $T_6 \in O(3^n)$
- $T_7 \in O(n^n)$
- $T_8 \in O(2^n)$

Then, $O(T_2(n)) \subset O(T_1(n)) \subset O(T_4(n)) \subset O(T_5(n)) \subset O(T_3(n)) \subset O(T_8(n)) \subset O(T_6(n)) \subset O(T_7(n))$

Approve by limit.

1. $$\lim_{n \to \infty} \frac{\log \log n}{\log n} = \frac{\infty}{\infty} \xrightarrow{\text{L'hopital}} \frac{\frac{1}{\ln(n) n \ln(2)}}{\frac{1}{n \ln 2}} = \frac{1}{\ln(n)}$$

$$\lim_{n \to \infty} \frac{1}{\ln(n)} = 0 \Rightarrow O(T_2(n)) \subset O(T_1(n))$$

2. $\lim_{n\to\infty} \frac{\log(n)}{n} = \frac{\infty}{\infty} \xrightarrow{\text{L'hospital}} \frac{\frac{1}{n\ln(2)}}{1} = \frac{1}{n\ln(n)}$

$\lim_{n\to\infty} \frac{1}{n\ln 2} = 0 \Rightarrow O(T_1(n)) \subset O(T_4(n))$

3. $\lim_{n\to\infty} \frac{n}{n^2} = \lim_{n\to\infty} \frac{1}{n} = 0 \Rightarrow O(T_4(n)) \subset O(T_5(n))$

4. $\lim_{n\to\infty} \frac{n^2}{n^3} = \lim_{n\to\infty} \frac{1}{n^3} = 0 \Rightarrow O(T_5(n)) \subset O(T_3(n))$

5. $\lim_{n\to\infty} \frac{n^5}{2^n} = \frac{\infty}{\infty} \xrightarrow{\text{L'hospital}} \frac{5n^4}{2^n\ln 2} \longrightarrow \frac{20n^3}{2^n\ln^2 2} \longrightarrow$

$\frac{60n^2}{2^n\ln^3 2} \longrightarrow \frac{120n}{2^n\ln^4 2} \longrightarrow \frac{120}{2^n\ln^5 2} \Rightarrow$

$\lim_{n\to\infty} \frac{120}{2^n\ln^5 2} = 0 \Rightarrow O(T_3(n)) \subset O(T_8(n))$

6. $\lim_{n\to\infty} \frac{2^n}{3^n} = \lim_{n\to\infty} \left(\frac{2}{3}\right)^n = 0 \Rightarrow O(T_7(n)) \subset O(T_6(n))$

7. $\lim_{n\to\infty} \frac{3^n}{n^n} = \lim_{n\to\infty} \left(\frac{3}{n}\right)^n = 0 \Rightarrow O(T_6(n)) \subset O(T_7(n))$

Q-2

a) $\lim_{n \to \infty} \frac{99n}{n} = 99$ (constant)

Therefore, $f(n) \in \Theta(g(n))$

b) $f(n) = 2n^4 + n^2 \in O(n^4)$

$\lim_{n \to \infty} \frac{n^4}{(\log n)^6} = \frac{\infty}{\infty} \xrightarrow{\text{L'hospital}} \frac{4n^3}{6(\log n)^5 \cdot \frac{1}{n}} \longrightarrow$

$\frac{12n^2}{30 \cdot \log^4 n \cdot \frac{1}{n} \cdot \frac{1}{n} + (-1 \cdot \frac{1}{n^2} \cdot 6\log^5 n)} \rightarrow \frac{12n^2}{\frac{1}{n^2}(30\log^4 n - 6\log^5 n)}$

$\frac{24n}{-2 \cdot \frac{1}{n^3} \cdot (30\log^4 n - 6\log^5 n) + \frac{1}{n^2} \cdot (120\log^3 n \cdot \frac{1}{n} - 30\log^4 n \cdot \frac{1}{n})}$

.... This limit goes to ∞ as the denominator of the denominator increases.

Therefore, $f(n) \in \Omega(g(n))$

c) $f(n) = \sum_{x=1}^{n} x = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$

Then $f(n) \in O\left(\frac{n^2}{2} + \frac{n}{2}\right) \in O(n^2)$

$g(n) \in O(4n + \log n) \in O(n)$

$$\lim_{n \to \infty} \frac{n^2}{n} = \lim_{n \to \infty} n = \infty$$

$$f(n) \in \Omega(g(n))$$

d) $$\lim_{n \to \infty} \frac{3^n}{5^{\sqrt{n}}} = \lim_{n \to \infty} \frac{3^{(\sqrt{n})^2}}{5^{\sqrt{n}}} = \lim_{n \to \infty} \left(\frac{3^{\sqrt{n}}}{5}\right)^{\sqrt{n}}$$

$$= \infty$$

$$f(n) \in \Omega(g(n))$$

Q-3 • MyFunction() returns the number which repeats more than "n/2" times until the n.th element of the array that took as parameter. Otherwise, if there is not such as element it returns "-1".

If inputs are: nums[] = {1, 1, 1, 2, 3, 4}

n = 5, then result is: 1

---

nums[] = {1, 1, 1, 2, 3, 4}

n = 6, then result is: -1

• The best case of algorithm is that "the first element will repeats more than $n/2$ times." In this case the inner loop will iterate $(n-1)$ times. And the basic operation (if(nums[j] == nums[i]) will be operated $(n-1)$ times.

Therefore $B(n) \in \Theta(n)$

- The worst case of this algorithm is that "not any element repeats more than $n/2$ times."

The basic operation (nums[j] == nums[i]) will be operate such as:

1st iteration $n-1$ time

2nd iteration $n-2$ time.

⋮

$(n-1)$th iteration 1 time.

It is $\sum_{i=1}^{n-1} i = (n-1)\cdot\frac{n}{2} = \frac{n^2-n}{2}$ times.

The worst case is $\Theta(n^2)$

Q-4 myFunction2 also returns the number that repeats more than n/2 times until n.th element of the array. If there is not such an element it returns -1.

- If inputs are: nums[] = {1, 1, 1, 2, 3, 4}
  n = 5, output = 1
  else if nums[] = {1, 1, 1, 2, 3, 4}
  n = 6, output = -1

- There are 3 loops inside the function. 2 of them iterates n times and not depends on data. One is depends data and iterates n time.

therefore, best and worst case are $\Theta(n)$

Q-5

	Time Complexities	
	Best	Worst
myFunction	$\Theta(n)$	$\Theta(n^2)$
myFunction2	$\Theta(n)$	$\Theta(n)$.

Since, myfunction2 is $\Theta(n)$ and myfunction has $O(n^2)$ in case of time complexity myfunction2 is better.

However, myFunction2 needs an map. Size of it cannot be predictable. In case of space complexity myFunction is much more better.

Q-6

a) $max_a = a_1$

for $i = 2$ to n

if a_i greater then max_a

set max_a to a_i

$max_b = b_1$

for $i = 2$ to m

if b_i greater then max_b

set max_b to b_i

Time complexity $W = B = \Theta(n) + \Theta(m)$

b) mergeSort (A)
mergeSort (B)
array C = sizeof (A + B).

merge 2 array

while elements on A finish OR elements on B finish.

if next element of A smaller then next element of B

add it to C.

else

add other element to C.

mergeSort (A) → $\Theta(n \log n)$

mergeSort (B) → $\Theta(m \log n)$

merging 2 array $\Theta(n) + \Theta(m)$

Time complexity is $\Theta(n \log n + m \log m)$.

c) if array is full.

create a double sized array

copy all elements to new array

add new element to the array. and assign it to prev. array.

else

add new element to array.

Time-complexity is amortized $O(1)$.

d) for i = 1 to n.

if element is not will be delete

add it to the new array.

else

pass the element and copy remain elements to the new array.

Time complexity is $\Theta(n)$.