Discussion 1

Dec. 16. 2020

1. Asymptotics & Limits

(a) prove: $n^3 = O(n^4)$

let $c=1$, for large n, $n^3 < n^4 \Leftrightarrow n > 1$

(b) Find $f(n), g(n) \geq 0$, s.t. $f(n) = O(g(n))$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq 0$

$f(n) = n$ $g(n) = 2n$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{1}{2}$ $f(n) = O(g(n))$

(c) prove: for any $c>0$, $\log n = O(n^c)$

~~Hint~~ use L'Hôpital's rule, if $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$

then $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}$

$\cancel{\lim_{n\to\infty} \frac{\log n}{n^c} = \lim_{n\to\infty} \frac{\frac{1}{n}}{c \cdot n^{c-1}} = \lim_{n\to\infty} \frac{1}{c\, n^c}}$

$\cancel{c>0 \Rightarrow \frac{1}{c \cdot n^c} \to}$

$\lim_{n\to\infty} \frac{\log n}{n^c} = \frac{1}{A} \frac{\frac{1}{n}}{c \cdot n^{c-1}} = \frac{1}{Ac} \lim_{n\to\infty} \frac{1}{n^c}$

$c>0 \Rightarrow \lim_{n\to\infty} \frac{1}{n^c} = 0 \Rightarrow f(n) = O(g(n))$

(d) Find an $f(n), g(n) \geq 0$ s.t. $f(n) = O(g(n))$, $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ doesn't exist.

(i) let $f(n) = \begin{cases} n & n \text{ is odd} \\ 2n & n \text{ is even} \end{cases}$ $g(n) = n$

for large n, $f(n) < g(n) \cdot 3 = 3n \Rightarrow f(n) = O(g(n))$ $(2n < 3n)$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n\ (\text{odd})\quad 2n\ (\text{odd})}{n}$ doesn't exist.

(ii) Another example

$f(n) = n(\sin n + 1)$ $g(n) = n$

let $c = 2$ $f(n) \leq 2g(n) = 2n \Rightarrow f(n) = O(g(n))$

$\lim_{n\to\infty} \frac{f(n)}{g(n)}$ doesn't exist

2. Asymptotic Notation Practice

(a)

(i) $f(n) = n$ $g(n) = n^2 - n \Rightarrow f(n) = O(g(n))$

ii) $f(n) = n^2 \quad g(n) = n^2+n \Rightarrow f(n) = \Theta(g(n))$

(iii) $f(n) = 8n \quad g(n) = n\log n \Rightarrow f(n) = O(g(n))$

iv $f(n) = 2^n \quad g(n) = n^2 \Rightarrow f(n) = \Omega(g(n))$

(v) $f(n) = 3^n \quad g(n) = 2^{2n} = 4^n \Rightarrow f(n) = O(g(n))$

(b) i) $f(n) = 50 \Rightarrow f(n) = \Theta(1)$

(ii) $f(n) = n^2 - 2n + 3 \Rightarrow f(n) = \Theta(n^2)$

(iii) $f(n) = n + \dots + 3 + 2 + 1 \Rightarrow f(n) = \Theta(n^2)$

(iv) $f(n) = n^{100} + 1.01^n \Rightarrow f(n) = \Theta(1.01^n)$

(v) $f(n) = n^{1.1} + n\log n \Rightarrow f(n) = \Theta(n^{1.1})$

(vi) $f(n) = (g(n))^2$ where $g(n) = \sqrt{n} + 5$

$f(n) = n + 10\sqrt{n} + 25 \Rightarrow f(n) = \Theta(n)$

3. Find the valley

An array A of length N. there's some j such that if $i<j$, $A[i] > A[i+1]$, if $i \geq j$, $A[i] \leq A[i+1]$, give an algorithm to find j.

Solution: Base Case: $N=1$, output 1

Induction: ① $A[N/2] < A[N/2+1]$ find $A[1 \dots N/2]$

② $A[N/2] > A[N/2+1]$ find $A[N/2+1 \dots N]$

$T(n) = O(\log n)$

Runtime & Correctness of Merge Sort

(a) Find base case and recurrence for $T(n)$.

$T(1) = 1 \quad T(n) = 2T(\frac{n}{2}) + c \cdot n$

(b) solve this recurrence.

$\log_b a = 1 = d \quad T(n) = O(n\log n)$

$$\begin{array}{ccccccc} & & n & & & & n \\ & \frac{n}{2} & & \frac{n}{2} & & & n \\ n/4 & n/4 & & n/4 & n/4 & & n \end{array} \Bigg\} \log n \text{ level}$$

(c) Consider the correctness. What's the desired property of C once it's completed. Required of the arguments to Merge for this?

Desire: C is sorted, Merge requires two arrays A and B are sorted.

(d) use the property of Merge, to show MergeSort is correct.

Induction: Base Case: $P(1)$ is true

for n, $P(k)$ is sorted $(k<n)$. (strong induction)

L and R are all sorted (length are all $< n$)

merge outputs a sorted array, which's the result.

5. Median of Medians

Quickselect (A, k)

(a) $A = [1, 2, \cdots n]$ shuffled into arbitrary order. QuickSelect (A, n/2) worst case?

The worst case is $1+2+\cdots+n = O(n^2)$

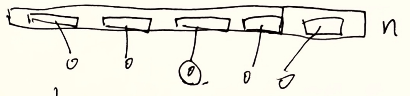
pivot: $1, 2, 3 \ldots \frac{n}{2}-1, n, n-1, \cdots \frac{n}{2}+1$

only have one element for one piece,

(b) A new algorithm Better-Quickselect picks a better pivot.

Split into n/5 then use median of medians as pivot.

Let p be the pivot, show that for at least $3n/10$ elements we have $p \geq x$, at least $3n/10$ elements, we have $p \leq x$.



Large or smaller: $\frac{n}{5} \times 3 \times \frac{1}{2} = \frac{3n}{10}$

(c) show that the worst case runtime of Better-QuickSelect (A, k) is $O(n)$

~~if~~ $\leq$

$$T(n) \leq T(n/5) + T(7n/10) + d \cdot n$$

for $m < n$, we have $T(m) < c \cdot n$

$$T(n) \leq \frac{cn}{5} + \frac{7cn}{10} + dn \leq cn$$

$$\frac{9c}{10} + d \leq c$$

$$c \geq 10d$$

then, $T(n) = O(n)$