

# Assignment 1

Zu Yu Zhang

## Problem 1

- $f_1(n) = 10^n \rightarrow$  exponential
- $f_2(n) = n^{\frac{1}{3}} \rightarrow$  polynomial
- $f_3(n) = n^n \rightarrow$  exponential
- $f_4(n) = \log_2 n \rightarrow$  logarithmic
- $f_5(n) = n \log_2 n \rightarrow$  poly-log

$$f_4 < f_2 < f_5 < f_1 < f_3$$

- logarithmic functions grow slower than polynomial functions, therefore  $f_4$  comes first, followed by  $f_2$  and  $f_5$ .
- $f_1$  and  $f_3$  are both exponential functions, let's compute the limit:

$$\lim_{n \rightarrow \infty} \frac{10^n}{n^n} = \lim_{n \rightarrow \infty} \left( \frac{10}{n} \right)^n = n \lim_{n \rightarrow \infty} \left( \frac{10}{n} \right) = n * 0 = 0$$

Therefore,  $10^n = O(n^n)$

## Problem 2

a.  $f(n) = n - 100 \quad g(n) = n - 200;$

$$f = \Theta(g)$$

- Constants can be ignored in asymptotic analysis; therefore, both  $f$  and  $g$  run in linear time.

b.  $f(n) = n^{\frac{1}{2}} \quad g(n) = n^{\frac{2}{3}}$

$$f = O(g)$$

- $\frac{2}{3} > \frac{1}{2}$ ;  $g$  grows faster than  $f$ .

c.  $f(n) = 100n + \log(n) \quad g(n) = n + (\log n)^2$

$$f = \Theta(g)$$

- $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 100$

- Since the limit converged to a constant (not zero),  $f$  and  $g$  grow at the same rate.

d.  $f(n) = n \log(n) \quad g(n) = 10n \log(10n)$

$$f = \Theta(g)$$

- Multiplicative constants can be ignored

e.  $f(n) = \log_2(n) \quad g(n) = \log_3(n)$

$$f = \Theta(g)$$

- Multiplicative constants can be ignored.

f.  $f(n) = 10 \log n \quad g(n) = \log(n^2)$

$$f = \Theta(g)$$

- According to logarithm product rule:
  - $\log(n)^2 = \log(n * n) = \log(n) + \log(n) = 2\log(n)$ ;
- Multiplicative constants can be ignored; f and g grow at the same rate.

g.  $f(n) = n^{1.01} \quad g(n) = n \log^2 n$

$$f = \Omega(g)$$

- $f(n) = n * n^{0.01}$
- $g(n) = n * (\log n) * (\log n)$
- Polynomials grows faster than logarithms.

h.  $f(n) = \frac{n^2}{\log n} \quad g(n) = n(\log n)^2$

$$f = \Omega(g)$$

- $\lim_{n \rightarrow \infty} \frac{n}{(\log n)^3}$
- polynomials grow faster than logarithmic functions, therefore this limit evaluates to  $\infty$  and f grows faster than g.

i.  $f(n) = n^{0.1} \quad g(n) = (\log n)^{10}$

$$f = \Omega(g)$$

- polynomials grow faster than logarithms.

j.  $f(n) = (\log n)^{\log n} \quad g(n) = \frac{n}{\log n}$

$$f = \Omega(g)$$

- $b^{\log_b(x)} = x$ , therefore  $b^{\log(\log n)} = \log n$
- $f(n) = \left(b^{\log(\log n)}\right)^{\log n} = (b^{\log n})^{\log(\log n)} = n^{\log(\log n)}$
- $\lim_{n \rightarrow \infty} \frac{n^{\log(\log n)}}{\frac{n}{\log n}} = \lim_{n \rightarrow \infty} n^{\log(\log n) - 1} * \log n$

$$= \lim_{n \rightarrow \infty} n^{\log(\log n) - 1} + \lim_{n \rightarrow \infty} \log n \text{ by product rule of limit laws}$$

$$= \infty + \infty$$

$$= \infty \rightarrow f \text{ grows faster than } g$$

$$k. \quad f(n) = \sqrt{n} \quad g(n) = (\log n)^3$$

$$f = \Omega(g)$$

- polynomials grow faster than logarithms.

$$l. \quad f(n) = n^{\frac{1}{2}} \quad g(n) = 5^{\log 2^N}$$

$$f = O(g)$$

- exponentials grows faster than polynomials.

$$m. \quad f(n) = n2^n \quad g(n) = 3^n$$

$$f(n) = O(g)$$

$$\bullet \quad \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$$

$$n. \quad f(n) = 2^n, \quad g(n) = 2^{n+1}$$

$$f = \Theta(g)$$

- $g(n) = 2^n * 2^1 \rightarrow g(n) = 2f(n)$
- f and g grow at the same rate since multiplicative constants are not considered.

### **Problem 3**

$$a. \quad T(n) = 2T\left(\frac{n}{3}\right) + 1$$

- $a = 2, b = 3, c = 0$
- $\log_3 2 > 0, T(n) = O\left(n^{\log_3(2)}\right)$

$$b. \quad T(n) = 5T\left(\frac{n}{4}\right) + n$$

- $a = 5, b = 4, c = 1$
- $\log_4 5 > 1, T(n) = O\left(n^{\log_4(5)}\right)$

$$c. \quad T(n) = 7T\left(\frac{n}{7}\right) + n$$

- $a = 7, b = 7, c = 1$
- $\log_7 7 = 1, T(n) = O(n \log n)$

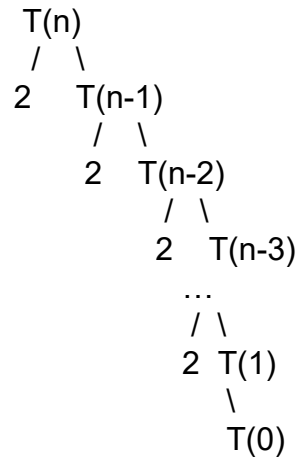
$$d. \quad T(n) = 9T\left(\frac{n}{3}\right) + n^2$$

- $a = 9, b = 3, c = 2$
- $\log_3 9 = 2, T(n) = O(n^2 \log n)$

e.  $T(n) = 8T\left(\frac{n}{2}\right) + n^3$

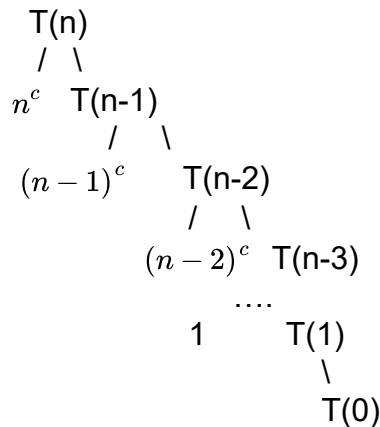
- $a = 8, b = 2, c = 3$
- $\log_2 8 = 3, T(n) = O(n^3 (\log n))$

g.  $T(n) = T(n-1) + 2$



- There is a total of  $n$  levels, on each level the work done is constant (2 units), the total work done is  $2n \rightarrow T(n) = O(n)$

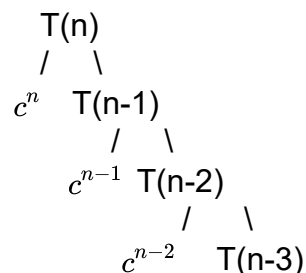
h.  $T(n) = T(n-1) + n^c$ , where  $c \geq 1$  is a constant



$$n^c + (n-1)^c + (n-2)^c + \dots + 1$$

- According to the Faulhaber's formula, this sum will be a polynomial function with degree  $(c+1)$ , therefore  $T(n) = O(n^{c+1})$

i.  $T(n) = T(n-1) + c^n$ , where  $c > 1$  is some constant



$$\begin{array}{c} \dots \\ 1 \quad T(1) \\ \quad \backslash \\ \quad T(0) \end{array}$$

$$c^n + c^{n-1} + c^{n-2} + \dots + c + 1$$

- Geometric sequence where the ratio is equal to c  $\rightarrow T(n) = O(c^{n+1})$ .

$$j) \quad T(n) = 2T(n-1) + 1$$

$$\begin{array}{ccccccc} & & T(n) & & & & \\ & / & & \backslash & & & \text{work done: 1} \\ T(n-1) & & & & T(n-1) & & \\ / & \backslash & & / & \backslash & & \text{work done: 2} \\ T(n-2) & T(n-2) & T(n-2) & T(n-2) & & & \\ \dots & \dots & \dots & \dots & & & \text{work done: 4} \\ T(1) & T(1) & & & T(1) & \dots & \end{array}$$

- sum of work done:
  - $2^0 + 2^1 + 2^2 + \dots + 2^n = 2^{n+1} - 1$
- Geometric sequence where the ratio is 2, sum will be exponential to 2, therefore  $T(n) = O(2^n)$ .

## References

[2.1.1 Recurrence Relation \(T\(n\)= T\(n-1\) + 1\) #1](#)

[https://en.wikipedia.org/wiki/List\\_of\\_logarithmic\\_identities](https://en.wikipedia.org/wiki/List_of_logarithmic_identities)

<https://math.stackexchange.com/questions/2605764/lognlogn-nlog10logn-why>

[https://www.ck12.org/book/cbse\\_maths\\_book\\_class\\_11/section/15.10/](https://www.ck12.org/book/cbse_maths_book_class_11/section/15.10/)