Lecture 5 - January 21

Asymptotic Analysis of Algorithms

From Absolute RT to Relative RT
Approximating RT Functions
Asymptotic Upper Bound (Big-O): Def.

Announcements/Reminders

- Assignment 1 released
- ***splitArrayHarder***: an extended version released
- Office Hours: 3pm to 4pm, Mon/Tue/Wed/Thu
- Contact Information of TAs on common eClass site

Example 2: Counting Number of Primitive Operations

(Exercise)

```
1 boolean foundEmptyString = false;
2 int i = 0;
3 while (!foundEmptyString && i < names.length) {
4   if (names[i].length() == 0) {
5     /* set flag for early exit */
6     foundEmptyString = true;
7   }
8   i = i + 1;
9 }
```

Q. # of times Line 3 is executed?

Q. # of times loop body (Lines 4 to 8) is executed?

Q. # of POs in the loop body (Lines 4 to 8)?

Comparing Algorithms: From Absolute RT to Relative RT

t

exact/absolute time taken to execute a P.O. in some exec. environment

e.g. Mac M1 $t = \boxed{2}$ (ms)

e.g. Mac'14 $t = \boxed{4}$ (ms)

abs. RT	Mac M1	Mac'14
$7n-2$	$(100 \cdot 7 - 2) \cdot 2$	$(100 \cdot 7 - 2) \cdot 4$
$10n+3$	$(100 \cdot 10 + 3) \cdot 2$	

absolute RTs of two alt. algorithms (solving the same problem)

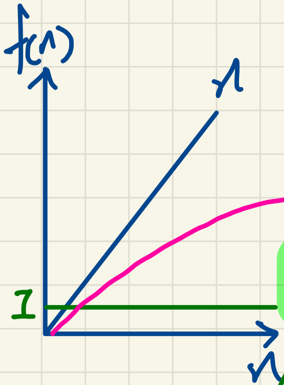
input size (assume $n = 100$)

when comparing two diff. alg.s not necessary to consider the extra, common factor. ↳ t discre.

when consid. the same alg., it's not necessary to know the precise RT difference

n^0 n^1 $\log n$

Exercise: Approximating $f(n) = 7n + 2n \cdot \log n + 3n^2$



by default, base 2

$$7n + 2n \cdot \boxed{\log} n + 3n^2$$

power: < 1

$$7 \cdot n^1 + 2 \cdot n^1 \cdot \boxed{\log_2 n} + 3 \cdot \boxed{n^2}$$

highest power (asymptotically, this highest term dominates over the rest)

$\approx n^{?}$ $\quad 0 < \_ < 1$

power: 1. ...

rate of growth faster than constant (n^0)

slower than linear (n^1)

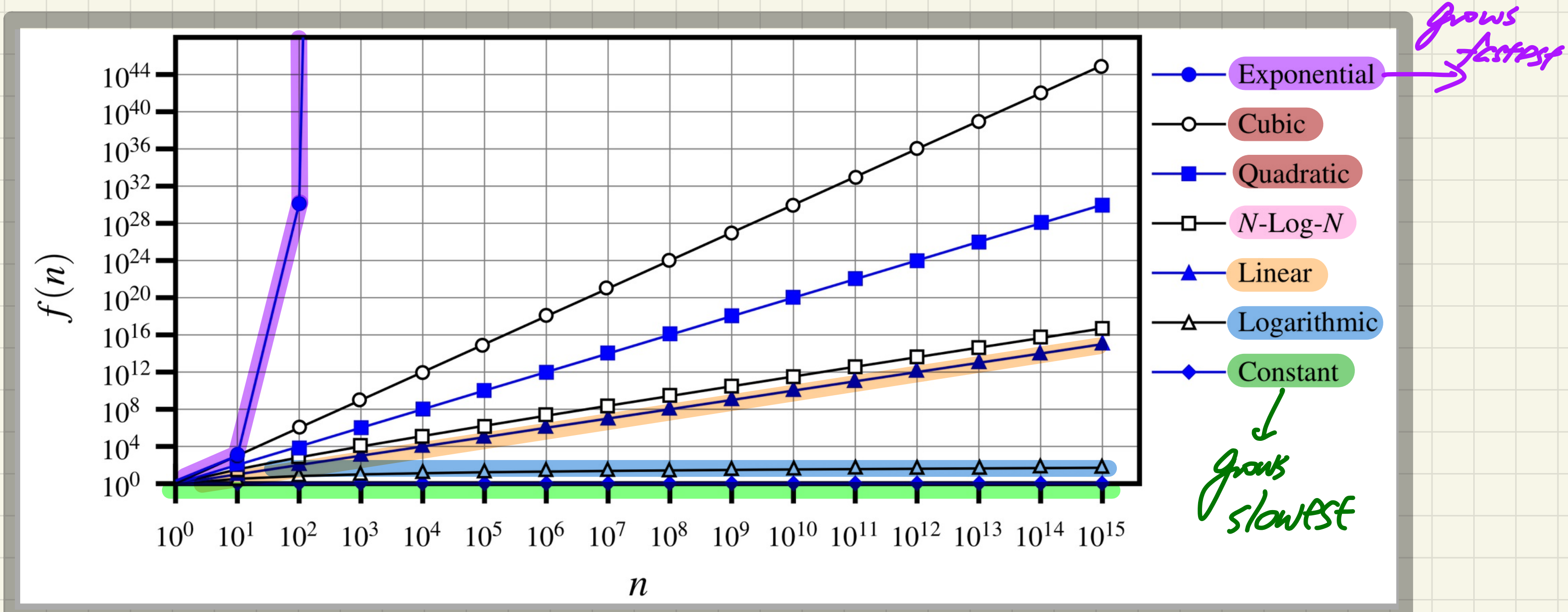
lower terms

multiplicative constants

Approximation:

n^2

RT Functions: Rates of Growth (w.r.t. Input Sizes)



RT

$f(n) = 1000 = 1000 \cdot n^{0}$

$f(1) = 1000$

$f(1000) = 1000$

$f(1M) = 1000$

Comparing Relative, Asymptotic RTs of Algorithms

Q1. Compare:

$$RT_1(n) = \cancel{3}n^2 + \cancel{7n} + \cancel{18} \approx n^2$$

$$RT_2(n) = 1\cancel{00}n^2 + \cancel{3n} - \cancel{100} \approx n^2$$

↳ equally efficient, asymptotically.

Q2: Compare:

$$RT_1(n) = n^3 + \cancel{7n} + \cancel{18} \approx n^3$$

$$RT_2(n) = 1\cancel{00}n^2 + \cancel{100n} + \cancel{2000} \approx n^2$$

↳ RT_2 more efficient (taking less time), asymptotically.

$f(n)$: RT function

↳ input size ⇝ relative RT

$f(n)$ → $O(g(n))$ e.g. $f(n) = 7n - 2$

$g(n)$: reference function

$g(n) \rightsquigarrow c \cdot g(n)$

(further manipulation on $g(n)$ expanded)

$\boxed{f(n)}$ being a member in the family means that it can somehow be upper-bounded by $\boxed{g(n)}$.

Goal: Prove $f(n)$ is $O(g(n))$

u.b.e. : upper-bound effect

Asymptotic Upper Bound: Big-O

$f(n) \in O(g(n))$ if there are:

- A real constant $c > 0$

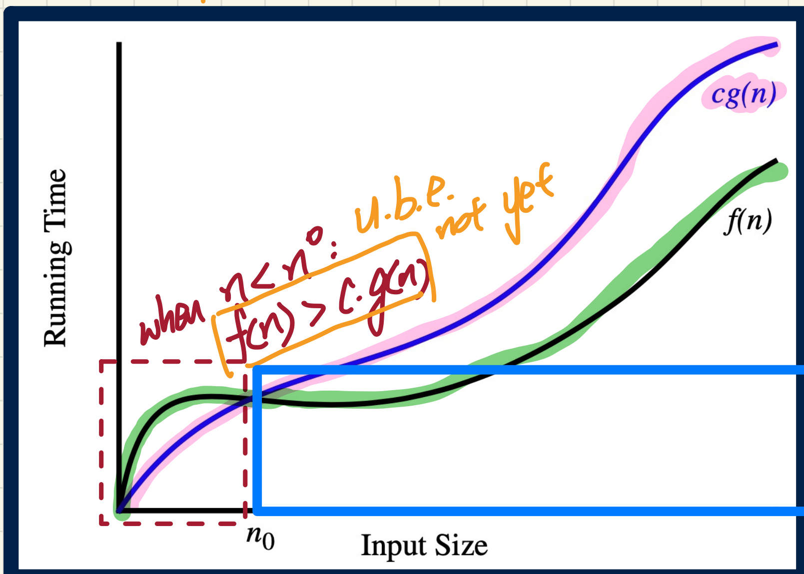
  multiplicative constant applied to g(n) to change its slope

- An integer constant $n_0 \geq 1$

  starting point of the u.b.e.

such that:

$$f(n) \leq c \cdot g(n) \quad \text{for } n \geq n_0$$



Example:

$f(\overset{1}{n}) = 8n + 5$

$g(n) = n$

Can n_0 be 1?

$f(1) \overset{?}{\leq} 9 \cdot n$

$8+5 = 13$

False

↳ u.b.e. not there.

Prove:

f(n) is O(g(n))

Choose c = 9.

What about n0?

→ starting from $n = n_0$

$f(n) \leq c \cdot g(n)$

u.b.e. is there!

$f(n) \le c \cdot g(n)$

Proving f(n) is O(g(n))

If $f(n)$ is a polynomial of degree d, i.e.,

$$f(n) = a_0 \cdot n^0 + a_1 \cdot n^1 + \cdots + a_d \cdot n^d$$

and $a_0, a_1, \ldots, a_d$ are integers (i.e., negative, zero, or positive), then $f(n)$ is $O(n^d)$. $\rightarrow g(n)$

We prove by choosing

$$\boxed{\begin{array}{l} c \\ n_0 \end{array}} \begin{array}{l} = |a_0| + |a_1| + \cdots + |a_d| \\ = 1 \end{array}$$

(1) $f(1) \le c \cdot \cancel{n}^d$ 1

(2) $f(n) \le c \cdot n^d$ $(n > 1)$

Upper-bound effect: $n_0 = 1$? $[f(1) \le (|a_0| + |a_1| + \cdots + |a_d|) \cdot 1^d]$

Upper-bound effect holds? $[f(n) \le (|a_0| + |a_1| + \cdots + |a_d|) \cdot n^d]$