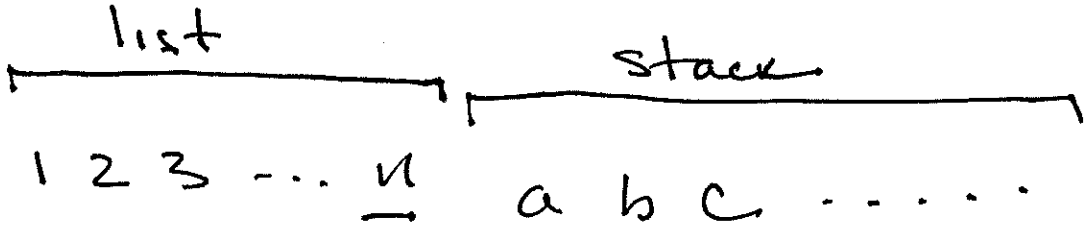
CSE 101-02 4-19-22

- midterm exam 1 : Thursday 4/21

  exam 65 min
  break 5 min
  lecture 25 min

- Pa4 : Portal.
- Pa3 : ext. 2 days, due Fri.

How to treat List S as a stack ?



Push = insertAfter

Handout: Algorithm Runtime

- iterative algorithms (only)
- Take CSE 102 for recursive algorithms.

What do we measure when we analyze runtime? count # of operations performed.

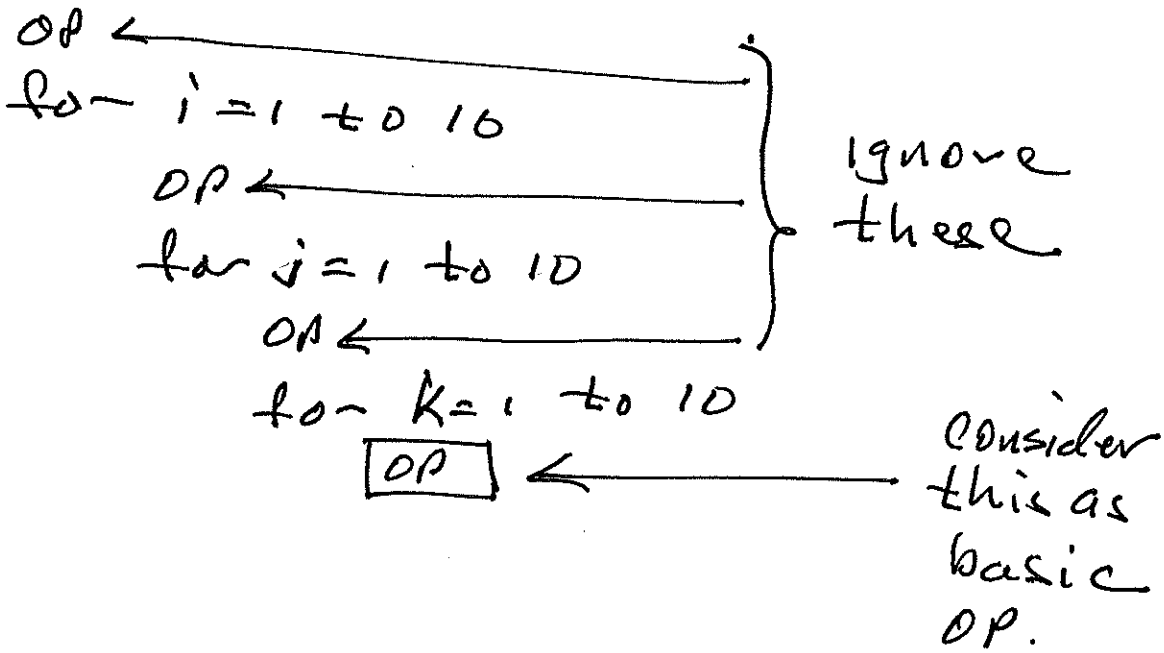
Ex.

```
OP1
for i = 1 to 10
    OP2
    for j = 1 to 10
        OP3
        for k = 1 to 10
            OP4
```

Operation	# executions
OP1	1
OP2	10
OP3	100
OP4	1000

- what if OP1 – OP4 are unequal cost?
- why not count all OPS?

Ex. Suppose OP1 = OP2 = OP3 = OP4 = OP



2 answers

$1 + 10 + 100 + 1000 = \underline{1,111 \text{ OPs}}$

Vs.

1000 OPs

why ignore OPs outside the innermost loop ?

Ex.

```
OP
for i = 1 to n
    OP
    for j = 1 to n
        OP
        for k = 1 to n
            OP
```

2 answers

$$n^3 + \underbrace{n^2 + n + 1}_{\text{ignore}} = T(n)$$

vs.

$\boxed{n^3}$

compare:

$$\lim_{n\to\infty}\left(\frac{T(n)}{n^3}\right) = \lim_{n\to\infty}\left(\frac{n^3+n^2+n+1}{n^3}\right)$$

$$= \lim_{n\to\infty}\left(1 + \frac{1}{n} + \frac{1}{n^2} + \frac{1}{n^3}\right) = 1$$

$\frac{1}{n} \to 0$, $\frac{1}{n^2} \to 0$, $\frac{1}{n^3} \to 0$

we would say $T(n)$ and n^3 are asymptotically equivalent.

Ex

```
for i = 1 to n
    for j = 1 to n
        for k = 1 to n
            OP
```

ans : $\boxed{n^3}$

Ex

```
for i = 1 to n
    for i = 1 to n
        for k = 1 to n
            OP
            OP
```

ans : $\boxed{2n^3}$

Assume these two do the same 'thing'.

We can equalize these by running 2nd on a machine that runs twice as fast.

So we don't distinguish between n^3 & $2n^3$.

Note: n^3 and n^2 cannot be equalized.

Informal Procedure:

1. choose a basic operation appearing inside the innermost loop.
2. count the # of executions as as a function of n = input size.

3. simplify function found in (2) by dropping lower order terms and replacing the lead coefficient by 1

Result: n^k for some k.

Actually could be more complicated.

Asymptotic Growth of functions

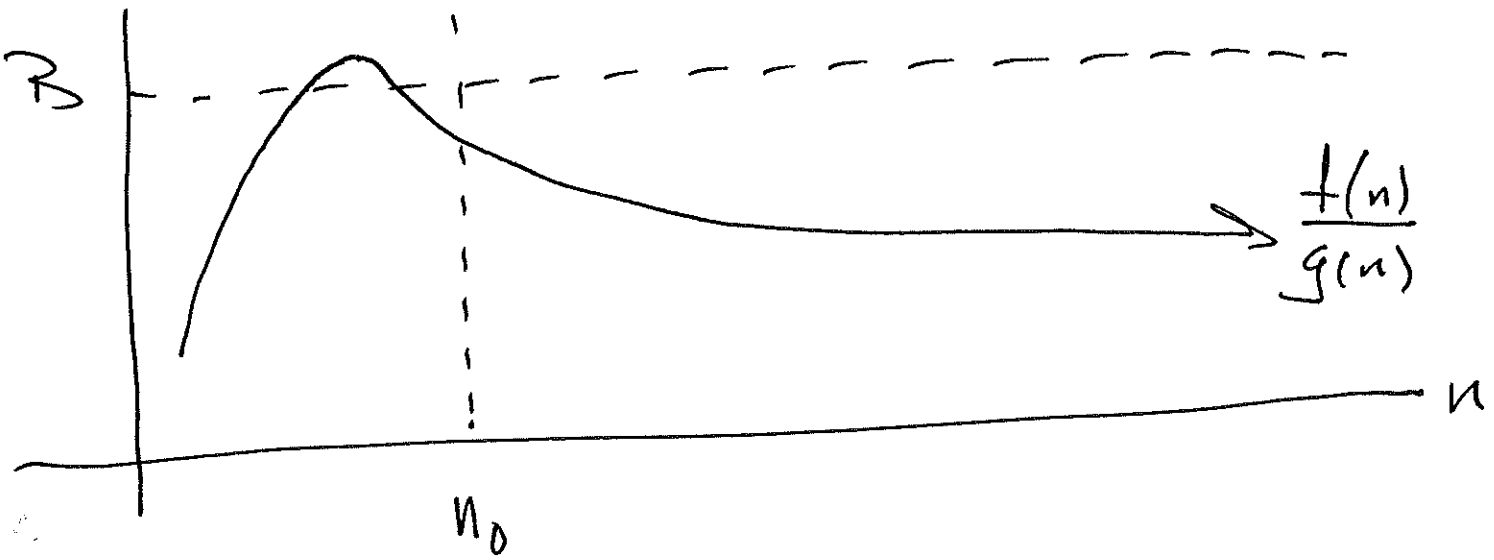
Let $f(n), g(n)$ denote positive functions.

Defn

write $f(n) = O(g(n))$ iff there exists $B > 0$ and $n_0 > 0$ such that

$$\frac{f(n)}{g(n)} \leq B \quad \text{for all } n \geq n_0 .$$

say "$g(n)$ is an asymptotic upper bound for $f(n)$" means: growth rate of $f(n)$ is no greater than that of $g(n)$.



note: B, n_0 are not unique.

Ex. $f(n) = 2n+5$, $g(n) = n$

$$\frac{f(n)}{g(n)} = 2 + \frac{5}{n} \leq 3 \quad \text{for} \quad n \geq 5$$

(B = 3, n_0 = 5)

Conclusion: $2n+5 = O(n)$.

note: nothing special about 2 & 5, i.e.

$$an+b = O(n)$$

for any $a > 0$, and any b

Hint: let $B = a+1$, $n_0 = \lceil |b| \rceil$

here $\lceil x \rceil$ is ceiling fcn.

Defn

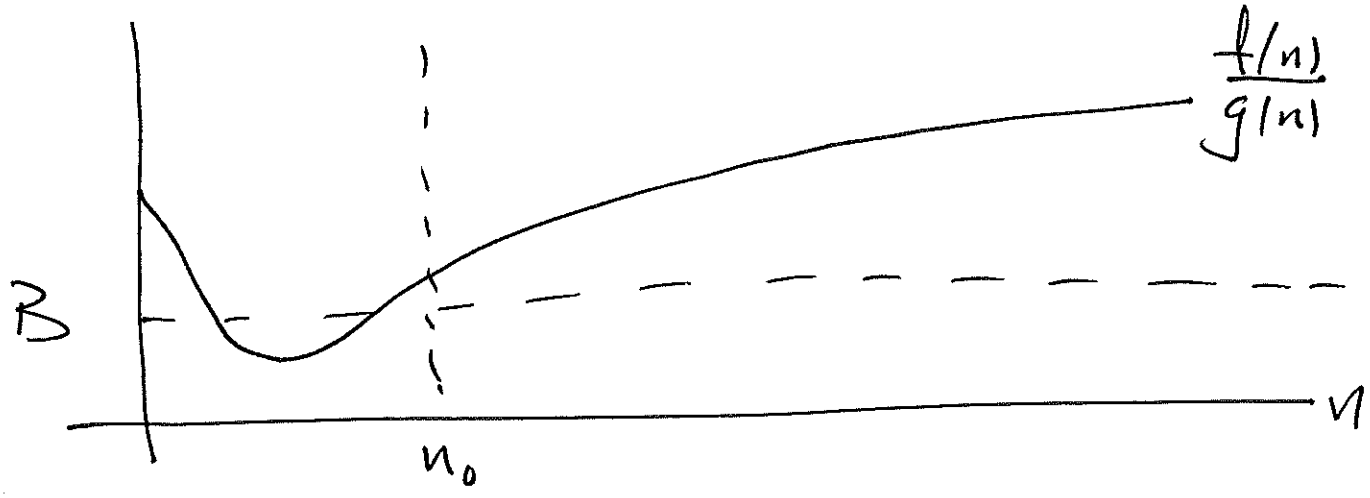
$f(n) = \Omega(g(n))$ iff there exist

$B > 0$ and $n_0 > 0$ such that

$$B \leq \frac{f(n)}{g(n)} \quad \text{for all } n \geq n_0$$

Meaning: $f(n)$ grows no slower than $g(n)$.

We say: $g(n)$ is an asymptotic lower bound for $f(n)$.



Ex.

Let $f(n) = 6n^3 + 4n$ and $g(n) = 2n^2$

observe

$$\frac{f(n)}{g(n)} = 3n + \frac{2}{n} \geq 7 \quad \text{for} \quad n \geq 2$$

$\uparrow$ B, $\uparrow$ n_0

conclusion: $6n^3 + 4n = \Omega(2n^2)$

observe: for any $a > 0$, b, c, d we have

$$an^3 + bn^2 + cn + d = \Omega(n^2)$$

Note:

$$\frac{f(n)}{g(n)} \leq B_1 \quad \text{iff} \quad \frac{g(n)}{f(n)} \geq \frac{1}{B_1} = B_2$$

so we see that

$$f(n) = O(g(n)) \quad \text{iff} \quad g(n) = \Omega(f(n)).$$

Thus, every O-statement is equivalent to a Ω-statement.

note: analogy: if $x, y \in \mathbb{R}$:

$$x \leq y \quad \text{iff} \quad y \geq x$$

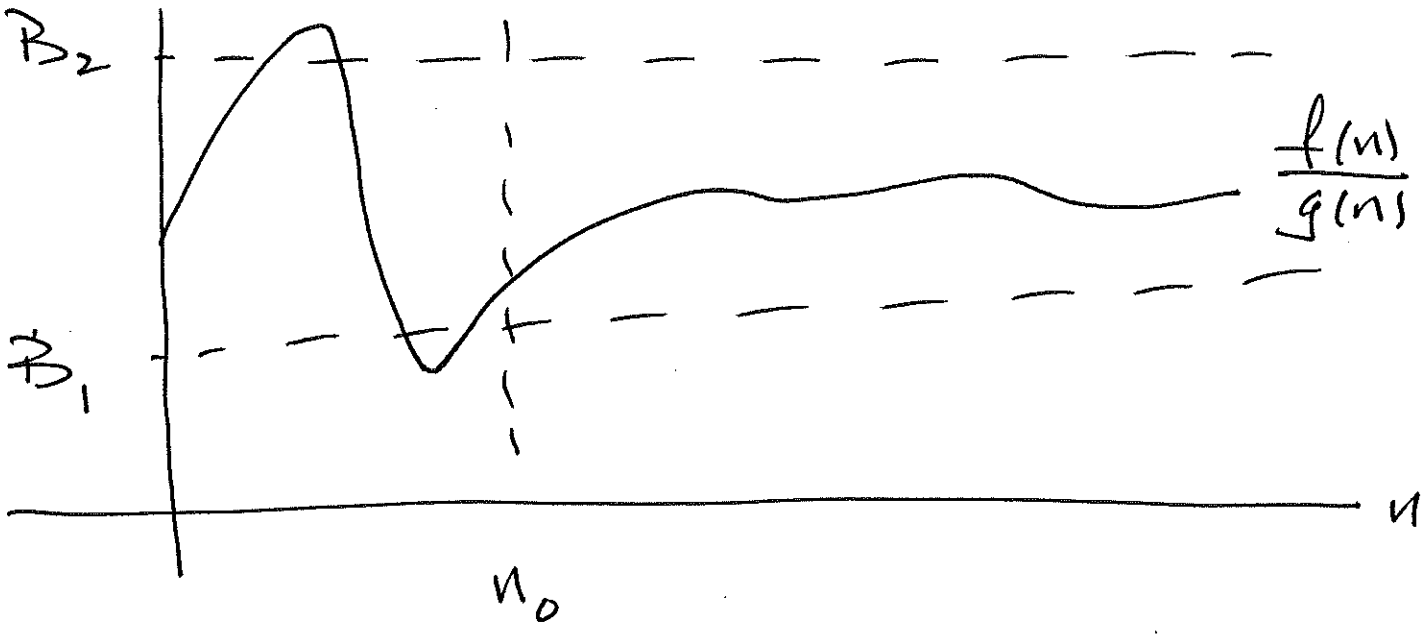
Defn

$f(n) = \Theta(g(n))$ iff there exist Positive B_1, B_2 and n_0 such that

$$B_1 \leq \frac{f(n)}{g(n)} \leq B_2 \quad \text{for all} \quad n \geq n_0$$

means: $f(n)$ and $g(n)$ grow at the same rate.

We say: $g(n)$ is a tight asymptotic bound on $f(n)$.



<u>observe</u>: $f(n) = \Theta(g(n))$ iff both

$f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

iff

$g(n) = \Omega(f(n))$ and $g(n) = O(f(n))$

iff

$$g(n) = \Theta(f(n)).$$