

# Lecture 9

Tuesday, September 22, 2020 5:00 PM

(Reminder: HW4 & lab1 are due this Sunday)

exam1: next Thursday

Example: We know that a laptop on average takes  $10^{-6}$  s to execute a single algorithm step (i.e. one line). When does the code finish if  $n=10^8$

① for  $i=1:n$   
    sum++  
end

$T(n) = \Theta(n)$   
exact  $(3n)$

# lines	time
1 line	$10^{-6}$ s
$3n = 3(10^8)$	$3(10^8) 10^{-6}$ s $\approx 3 \times 10^{8-6}$ s $\approx 300$ s

② for  $i=1:n^2$   
    for  $j=1:i$   
        sum++  
    end  
end

$$T(n) = \sum_{i=1}^{n^2} \sum_{j=1}^i 1 = \sum_{i=1}^{n^2} i = \frac{n^2(n^2+1)}{2} \approx \Theta(n^4)$$

$$\Theta\left(\int_1^{n^2} x dx\right) \approx \Theta(n^4)$$

1 line  $\rightarrow 10^{-6}$  s  
 $n^4 \rightarrow n^4 10^{-6} = (10^8)^4 10^{-6}$  s  
 $= 10^{32-6} = 10^{26}$  s

$\frac{10^{26}}{60 \times 60} \rightarrow \frac{10^{26}}{100 \times 100} = \frac{10^{26}}{10^4} = 10^{22}$  hrs

③ for  $i=1:\log n$

    for  $j=1:i$   
        sum++  
    end  
end

$$T(n) = \sum_{i=1}^{\log n} \sum_{j=1}^i 1 = \sum_{i=1}^{\log n} i = \frac{\log n (\log n + 1)}{2} \approx \Theta(\log^2 n)$$

$$\Theta\left(\int_1^{\log n} x dx\right) \approx \Theta(\log^2 n)$$

1 line  $\rightarrow 10^{-6}$  sec

$\log^2 n \rightarrow 10^{-6} \log^2 10^8$  s  $= 8 \times 10^{-6} \log^2 10$

$= 64 \times 10^{-6} \times (3 \sim)^2$

$\approx 64 \times 10^{-6} \times 9 > 64 \times 10^{-6} \times 10$

$64 \times 10^{-5} = 6.4 \mu s$

end

$$\approx 64 \times 10^0 \times 9 > 64 \times 10 \times 10^5$$

$$64 \times 10^5 = \underline{6.4 \mu s}$$

④ for i=1:n  
for j=1:i<sup>2</sup>

for k=1:j

sum++

end

end

end

$$T(n) = \sum_{i=1}^n \sum_{j=1}^{i^2} \sum_{k=1}^j 1 = \sum_{i=1}^n \sum_{j=1}^{i^2} j = \sum_{i=1}^n \frac{i^2(i^2+1)}{2}$$

$$\approx \sum_{i=1}^n i^4 = \Theta\left(\sum_{i=1}^n i^4 dx\right) = \Theta(n^5)$$

$$1 \text{ line} \rightarrow 10^{-6} s$$

$$n^5 \rightarrow n^5 10^{-6} = (10^8)^5 10^{-6} s$$

$$= 10^{34} s = \frac{10^{34}}{60 \times 60} > \frac{10^{34}}{10^4} \approx 10^{30} \text{ hrs}$$

⑤\*

for i=1:n

for j=1:S[i]

sum++

end

end

info: S = [5, 6, 1, 10, 2, ...]  $\Rightarrow |S| = n$

$$T(n) = \sum_{i=1}^n \sum_{j=1}^{S[i]} 1 = \sum_{i=1}^n S[i]$$

$$= n^3 = \Theta(n^3)$$

$$1 \text{ line} \rightarrow 10^{-6} \text{ sec}$$

$$n^3 \text{ lines} \rightarrow 10^{-6} n^3 = 10^{-6} (10^8)^3 = 10^{24-6} = 10^{18} \text{ sec}$$

⑥\*

for i=1:n<sup>2</sup>

if i < log n

sum++

else

break

end

end

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

$$1 \text{ line} \rightarrow 10^{-6} \text{ sec}$$

$$\log n \rightarrow 10^{-6} \log n = 10^{-6} \log 10^8 s$$

$$= 8 \times 10^{-6} \times \log 10 \approx 24 \times 10^{-6} s$$

$$\approx 24 \mu s$$

⑦\*

for i=1:n

info: f(n) = \Theta(n), g(n) = \Theta(n^2)

even  $\rightarrow 2k$   
odd  $\rightarrow 2k+1$

⑦\*  
for  $i = 1 : n$

info:  $f(n) = \Theta(n)$ ,  $g(n) = \Theta(n^2)$

even  $\rightarrow 2k$   
odd  $\rightarrow 2k+1$

if  $i \% 2 \rightarrow \text{odd}$

Sum += f(i)

else  $\rightarrow \text{even}$

Sum += g(i)

end

end

$$T(n) = \Theta(n^3)$$

$$T(n) = \sum_{\text{odd}} f(i) + \sum_{\text{even}} g(i)$$

$$= \sum_{i=\text{even}} i^2 = \sum_{k=1}^{n/2} (2k)^2$$

$$= \sum_{k=1}^{n/2} 4k^2 = \Theta\left(\int_1^{n/2} x^2 dx\right) = \Theta(n^3)$$

## Linear Search

$a = [10 \ 5 \ 2 \ 0 \ 4 \ 6] \rightarrow |a| = n$

key = 10

Best-Case:  $\Omega(1)$

key = 8  
Worst-case:  $O(n)$

key = 2

avg-case:  $O(n)$

Linear\_Search(a, key)

$n = a.length$

for  $i = 1 : n$

if  $a[i] == key$

return True;

end

end

return False;

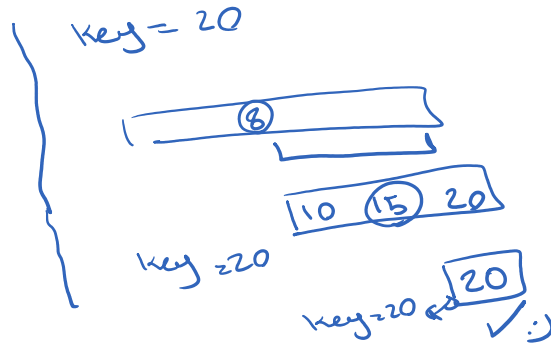
$$T(n) = O(n)$$

Binary-Search: Only works for sorted arrays

$a = [2 \ 3 \ 5 \ 8 \ 10 \ 15 \ 20]$

key = 8

best-case:  $\Omega(1)$



<u>step</u>	<u>Size</u>	<u>amt work</u>
→ 0	$\frac{n}{2^0}$	C
1	$\frac{n}{2^1}$	C
2	$\frac{n}{2^2}$	C
3	$\frac{n}{2^3}$	C
⋮	⋮	⋮
k	$1 = \frac{n}{2^k}$	C
$n = 2^k$ $\log n = k$		$T(n) \leq C + C + C + C + \dots + C$ $\leq C(k+1)$ $T(n) = O(k(k+1))$ $= O(k^2)$ $= O(k)$ $= O(\log n)$ ✓

## Part B Lab 1

① run BS once

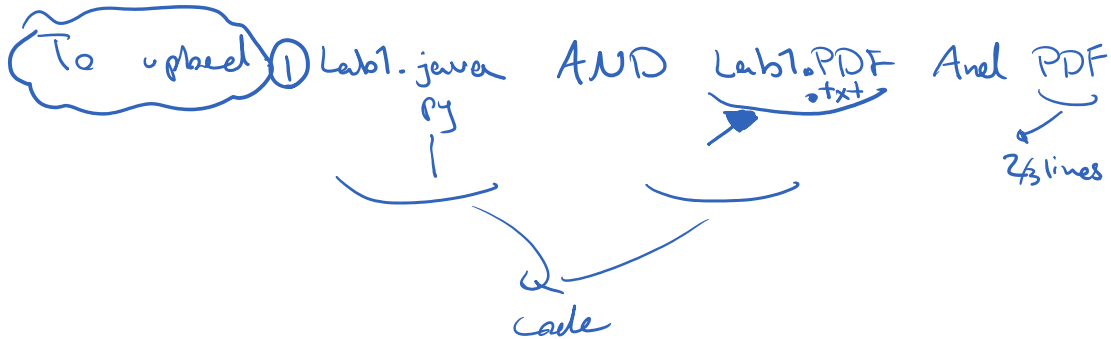
it takes to run 1 line

① run BS once

② estimate how much it takes to run 1 line using the time complexity of BS.

③ Using 2 (like example 1 today) estimate when you finish running the code (BS or LS)

write a code, it should be ~2 lines



② I don't need to see the output