Lecture 9

Tuesday, September 22, 2020 5:00 PM

(Reminder: HW4 & lab7 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

$i=1,2,\dots,n$

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

exact: $3n$

# lines	time
1 line	$10^{-6}s$
$3n = 3(10^8)$	$3(10^8)\,10^{-6}s = 3\times10^{8-6}s = \boxed{300s}$

② for i=1:n^2

for j=1:i

sum++

end

end

$$T(n) = \sum_{i=1}^{n^2}\sum_{j=1}^{i} 1 = \sum_{i=1}^{n^2} i = \begin{cases} \frac{n^2(n^2+1)}{2} = \Theta(n^4) \\ \Theta\left(\int^{n^2} x\,dx\right) = \Theta(n^4) \end{cases}$$

1 line $\to 10^{-6}s$

$n^4 \to n^4\,10^{-6} = (10^8)^4\,10^{-6}s$

$= 10^{32-6} = \boxed{10^{26}s}$ ✓

hrs $\rightsquigarrow \frac{10^{26}}{\underbrace{60}_{mins}\times\underbrace{60}_{hrs}} > \frac{10^{26}}{100\times100} = \frac{10^{26}}{10^4} = \boxed{10^{22}\,hrs}$

③ for i=1:logn

for j=1:i

sum++

end

end

$$T(n) = \sum_{i=1}^{\log n}\sum_{j=1}^{i} 1 = \sum_{i=1}^{\log n} i = \begin{cases} \frac{\log n(\log n+1)}{2} = \Theta(\log^2 n) \\ \Theta\left(\int_1^{\log n} x\,dx\right) = \Theta(\log^2 n) \end{cases}$$

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

$\log^2 n \to 10^{-6}\log^2 10^8\,s = 8^2\times10^{-6}\log^2 10$

$= 64\times10^{-6}\times(3.\sim)^2$

$\simeq 64\times10^{-6}\times9 > 64\times10^{-6}\times10\,s$

$64\times10^{-5} = \boxed{6.4\,\mu s}$

end

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

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

---

(4)
```
for i=1:n
    for j=1:i^2
        for k=1:j
            sum++
        end
    end
end
```

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

$$\simeq \sum_{i=1}^{n} i^4 = \Theta\left(\int_1^n x^4 dx\right) = \boxed{\Theta(n^5)}$$

1 line $\longrightarrow 10^{-6}$ s

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

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

---

(5) *
```
for i=1:n
    for j=1:S[i]
        sum++
    end
end
```

info: $S = [5, 6, 1, 10, 2, \text{---}] \Rightarrow |S| = n$

$\Rightarrow \sum_{i=1}^{n} S[i] = n^3$

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

$$= n^3 = \boxed{\Theta(n^3)}$$

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

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

---

(6) *
```
for i=1:n^2
    if i < logn
        sum++
    else
        break
    end
end
```

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

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

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

$= 8 \times 10^{-6} \times \log 10 \simeq 24.\text{-} \times 10^{-6}$ s

$\log 10 \to 3.\text{-}$

$\boxed{\simeq 24\ \mu s}$

---

(7) *
```
for i=1:n
```

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

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

7 *

```
for i=1:n
    if i%2  ~> odd
        sum += f(i)
    else  -> even
        sum += g(i)
    end
end
```

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

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

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

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

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

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

Linear Search

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

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;
```

$\boxed{T(n) = O(n)}$

Binary-Search: Only works for sorted arrays

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

key = 8

best-case : $\Omega(1)$

key = 20

8

10 15 20

key = 20

20

key = 20

✓ :)

step	size	amt work
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

$n = 2^k$

$\log n = k$

$T(n) \leq c + c + c + c + \cdots + c$

$\leq c(k+1)$

$T(n) = O(\cancel{c}(k+1))$

$= O(k + \cancel{1})$

$= O(k)$

$\boxed{= O(\log n)}$ ✓

Part B Lab7

(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

To upload ① Lab1.java AND Lab1.PDF And PDF
py .txt
code
2/3 lines

② I don't need to see the output