2110681 Computer Algorithms: (Lecture) Example I

Ex: Find Max1

```
001     begin
002     answer ← A[1]
003             for i ← 2 step 1 until n do
004                     if A[i] > answer then answer = A[i] endif
005             enddo
006     return answer
007     end
```

Solution:

Time used : 002 → 1
003 → n
004 → n-1
005 → 1
006 → 1

That is, $T_A(n) = T_{002} + nT_{003} + (n-1)T_{004} + T_{005} + T_{006}$

$= n + n - 1 + 3$

$= 2n+2$

#

Ex: Find Max2

```
001     max-search(A[1...n])
002     begin
003             if n = 1 then return A[1] endif
004             answer ← max-search(A[1...n-1])
005             if A[n] > answer then return A[n] else return answer endif
006     end
```

Solution:

Time used: $T_A(n) = T_{003} + T_{004} + T_A(n-1) + T_{005}$

$= T_{003} + T_{004} + T_{005} + (T_{003} + T_{004} + T_{005} + T_A(n-2)) + ... + T_{003}$

$= nT_{003} + (n-1)T_{004} + (n-1)T_{005}$

$= n + (n-1) + (n-1)$

$= 3n+2$

#

Ex: Bubble Sort

```
001     sort(A[1…n])
002     begin
003     for last ← n step -1 until 2 do
004             for i ← 2 step 1 until last do
005                     if A[i] < A[i-1] then   buffer ← A[i]
                                                A[i] ← A[i-1]
                                                A[i-1] ← buffer endif
006             enddo
007     enddo
008     end
```

Solution:

Time used : $T_A(n) = (n-1) + (n-2) + (n-3) + \ldots + 1 = n(n-1)/2$

That is $T_A(n) = \Theta(n^2)$

#

Ex: Insertion Sort

```
sort(A[1…n])
begin
        for j ← 2 step 1 until n do
                buffer ← A[j]
                i ← j-1
                while i > 0 and A[i] > buffer do
                        A[i+1] ← A[i]
                        i ← i-1
                enddo
                A[i+1] ← buffer
        enddo
end
```

Solution:

Time used: $T_A(n) = 1 + 2 + 3 + \ldots + n = n(n-1)/2$

Worst case: $O(n^2)$

Best case: $\Omega(n)$

#

Ex: Find time complexity of

```
for i ← 1 to n do
        …
        y ← S(m)
        …
enddo
```

if $S(m) = \Theta(m^2)$.

Solution:

$$T(n) = \sum_{i=1}^{n} \Theta(m^2) = \Theta(m^2)\sum_{i=1}^{n} 1$$

$$= n\Theta(m^2) = \Theta(nm^2)$$

#

Ex: Find time complexity of

```
for i ← 1 to n do
        …
        y ← S(i)
        …
enddo
```

if $S(i) = \Theta(n^2)$.

Solution:

$$T(n) = \sum_{i=1}^{n} \Theta(i^2) = \Theta(\sum_{i=1}^{n} i^2) = \Theta(n^3)$$

#

Ex: Find time complexity of

001 for $i \leftarrow 1$ to n do
  …
002   for $j \leftarrow 1$ to i do
    …
003   S(j)
004   enddo
005 enddo

if S(j) = $\Theta(n^2)$.

Solution:

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

$$= \Theta(\sum_{i=1}^{n} i^3) = \Theta(n^4)$$

#

Ex: Find time complexity of

001 for $i_1 \leftarrow 1$ to n do
002   for $i_2 \leftarrow 1$ to i_1 do
003   for $i_3 \leftarrow 1$ to i_2 do
      …
00m for $i_m \leftarrow 1$ to i_{m-1} do
00m+1   S ← S + 1

if S = 0.

Solution:

001 → $1 \le i_1 \le n$
002 → $1 \le i_2 \le i_1 \le n$
003 → $1 \le i_3 \le i_2 \le i_1 \le n$
…
00m → $1 \le i_m \le i_{m-1} \le \ldots \le i_2 \le i_1 \le n$

```
(1)     (2)    (3)      ...    (n-1)   (n)
 x   |  xx  |  x    |       |        |
 0   1  00  1  0    1       1        1
```

Since #0 = m

#1 = n-1, so # of bit string = m+n-1.

We can conclude that $T_S = C_m^{m+n-1} = C_{n-1}^{m+n-1}$

#

Ex:

001 while $n > 0$ do
002   $n \leftarrow \lfloor n/2 \rfloor$
003 enddo

Solution:

Iteration:	1	2	3	4	…	k-1	k
	n	$n/2$	$n/2^2$	$n/2^3$		$\lfloor n/2^{k-2} \rfloor$	$0 = \lfloor n/2^{k-1} \rfloor$

Thus, $2^{k-2} \le n < 2^{k-1}$

$\log_2 2^{k-2} \le \log_2 n < \log_2 2^{k-1}$

$k-2 \le \log_2 n < k-1 < k$

$k \sim \log_2 n$

That is $T(n) = \Theta(\log_2 n)$

#

Ex: Find the time complexity of GCD(a, b)

```
begin
while (a > 0) do
        q ← a
        a ← b mod a
        b ← q
enddo
return b
end
```

Solution:

Since $b \bmod a = b - \lfloor b/a \rfloor a$, that is $0 \le b \bmod a \le a - 1$

case 1: $a < b/2$, $b \bmod a \le a-1 < b/2$

case 2: $a = b/2$, $b \bmod a = 0$

case 3: $a > b/2$, $b/2 < a < b$. Since $b/2 < a$

$b/a < 2$

and $a < b$

$b/a > 1$, that is $1 < b/a < 2$.

From this, we can conclude that $\lfloor b/a \rfloor = 1$.

So, $b \bmod a = b - a < b - b/2 = b/2$

The value of a in each loop. For the i^{th} iteration, approximate by $b/2$.

Since the number of iterations is equal to $\log_2 b$, time complexity = $O(\log b)$

#

Ex: $f_n = f_{n-1} + f_{n-2}$ where $f_0 = 0$, $f_1 = 1$

Solution:

χ: $r^2 - r - 1 = 0$

$$r_1 = \frac{1+\sqrt{5}}{2}, \; r_2 = \frac{1-\sqrt{5}}{2}$$

Thus, $f_n = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^n$

$f_0 = 0 = \alpha_1 + \alpha_2 \rightarrow \alpha_1 = -\alpha_2$

$f_1 = 1 = \alpha_1 \frac{1+\sqrt{5}}{2} + \alpha_2 \frac{1-\sqrt{5}}{2}$

$= \alpha_1 \frac{1+\sqrt{5}}{2} - \alpha_1 \frac{1-\sqrt{5}}{2}$, $\alpha_1 = \frac{1}{\sqrt{5}}$ and $\alpha_2 = -\frac{1}{\sqrt{5}}$

That is, $f_n = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n - \frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})^n$

Test: $f_0 = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^0 - \frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})^0 = 0$

$f_1 = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^1 - \frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})^1 = 1$

$f_2 = \ldots$

#

Ex: Find the time complexity of Insertion Sort, $T(n) = T(n-1) + n$, $T_1 = 0$
Solution: $T_n = T_{n-1} + T_n$

Homogeneous part: $T_n^{(h)} = T_{n-1}$

χ: r = 1

Therefore, $T_n^{(h)} = \alpha_1(1)^n$ ....(1)

Particular part: $f(n) = n$

$$T_n^{(p)} = (p_1 n + p_2)n$$
$$p_1 n^2 + p_2 n = p_1(n-1)^2 + p_2(n-1) + n$$
$$= p_1(n^2-2n+1) + p_2(n-1) + n$$
$$= p_1 n^2 - 2p_1 n + p_1 + p_2 n - p_2 + n$$
$$p_1 n^2 + p_2 n = p_1 n^2 + (-2p_1 + p_2 + 1)n + p_1 - p_2$$

Then, $p_2 = -2p_1 + p_2 + 1$

$p_1 = 1/2$

and $0 = p_1 - p_2$,

$p_2 = p_1 = 1/2$

That is, $T_n^{(p)} = (½)n^2 + (½)n$ ....(2)

From (1) and (2), $T_n = \alpha_1(1)^n + (½)n^2 + (½)n$

Find α_1, $T_1 = 0 = \alpha_1 + 1 \rightarrow \alpha_1 = -1$

Hence, $T_n = -1(1)^n + (½)n^2 + (½)n$

$= -1 + (½)n^2 + (½)n$

$= \frac{n(n+1)}{2} - 1$

$T_n = O(n^2)$

#

Ex: Find the time complexity of Binary Search, $T_n = T_{n/2} + 1$, $T_1 = 1$
Solution:

Let $T_b = T_{b-1} + 1$, where $b = \log_2 n$.

Homogeneous part: $T_b^{(h)} = \alpha_1(1)^b$ ....(1)

Particular part: $T_b^{(p)} = p_1 b$

$p_1 b = p_1(b-1) + 1$

$0 = -p_1 + 1$, $p_1 = 1$

So, $T_b^{(p)} = b$ …………(2)

From (1) and (2), $T_b = \alpha_1(1)^b + b$

Find α_1: $T_{n=1} = 1$, $b = \log_2 n$

$= \log_2 1 = 0$, that is $T_{b=0} = 1$.

Wherefore, $T_0 = 1$

$= \alpha_1 + 0 = 1$, $\alpha_1 = 1$.

We can conclude that, $T_b = (1)^b + b$.

Because of $b = \log_2 n$,

thus, $T_b = \log_2 n + 1 = O(\log n)$

#

Ex: Merge Sort, $T_n = 2T_{n/2} + n$, $T_1 = 0$

Solution:

Let $T_b = 2T_{b-1} + 2^b$, where $b = \log_2 n$.

Homogeneous part: $T_b^{(h)} = \alpha_1(2)^b$ …………(1)

Particular part: $T_b^{(p)} = p_1 2^b b$

$p_1 2^b b = 2p_1 2^{b-1}(b-1) + 2^b$

$p_1 b = p_1(b-1) + 1$

$p1 = 1$

That is, $T_b^{(p)} = 2^b b$ …………(2)

(1) and (2), $T_b = \alpha_1(2)^b + 2^b b$

Find α_1: $T_{n=1} = 0$. Thus $T_{b=0} = 0$, since $b = \log_2 n = \log_2 1 = 0$.

Thence, $T_0 = 0 = \alpha_1 \rightarrow \alpha_1 = 0$.

We can conclude that $T_b = 2^b b$

Replace b with $\log_2 n$, $Tn = (2^{\log_2 n})\log_2 n = n \log_2 n$

$= \Theta(n \log n)$

#

Ex: $T_n = 2T_{n/2} + n^2$, $T_1 = 0$

Solution:

Let $T_b = 2T_{b-1} + 2^{2b}$, where $b = \log_2 n$.

Homogeneous part: $T_b^{(h)} = \alpha_1(2)^b$ …………(1)

Particular part: $T_b^{(p)} = p_1 2^{2b} b$

$p_1 2^{2b} b = 2p_1 2^{2(b-1)}(b-1) + 2^{2b}$

$= p_1 2^{2b-1)}(b-1) + 2^{2b}$

$p_1 b = p_1 b/2 - p_1/2 + 1$

$p_1 b = -p_1 + 2$

$p_1 = \frac{2}{b+1}$

So, $T_b^{(p)} = \frac{2}{b+1} 2^{2b} b$ …………(2)

From (1) and (2), $T_b = \alpha_1(2)^b + \frac{2}{b+1} 2^{2b} b$

Since $T_{n=1} = 0$, $T_{b=0} = 0$.

That is, $T_{b=0} = \alpha_1 + 0 = 0. \rightarrow \alpha_1 = 0$.

Therefore, $T_b = \frac{2}{b+1} 2^{2b} b$,

$$T_n = \frac{2}{(\log_2 n)+1} 2^{2\log_2 n} \log_2 n$$

$$= \frac{2}{(\log_2 n)+1} n^2 \log_2 n \qquad < \frac{2}{\log_2 n} n^2 \log_2 n$$

$$= 2n^2$$

$$= O(n^2)$$

#