



# CSC240

## Lecture 11 Week 8

Author: Joseph Siu

Email: [joseph.siu@mail.utoronto.ca](mailto:joseph.siu@mail.utoronto.ca)

Date: March 14, 2024



For  $n \in \mathbb{Z}^+$ , let  $M(n) = \begin{cases} c & \text{if } n = 1 \\ M(\lfloor \frac{n}{2} \rfloor) + M(\lfloor \frac{n}{2} \rfloor) + dn & \text{if } n > 1 \end{cases}$ .

For all  $i \in \mathbb{N}$ , let  $Q(i) = "M(2^i) = c2^i + di2^i"$ .

### Claim 1

$\forall i \in \mathbb{N}. Q(i)$ .



### Theorem 1

$M(n) \in \Theta(n \log n)$ .



### Lemma 1

$\forall n \in \mathbb{Z}^+. \forall m \in \mathbb{Z}^+. (m \leq n) \text{ IMPLIES } (M(m) \leq qM(n))$ .

That is,  $M$  is a non-decreasing function.



*Proof of Lemma 1 by induction on  $n$ .*

For  $n \in \mathbb{Z}^+$ . Let  $R(n) = "\forall m \in \mathbb{Z}. (m \leq n) \text{ IMPLIES } (M(m) \leq M(n))"$

Let  $n \in \mathbb{Z}^+$  be arbitrary, suppose  $R(n')$  is true for all  $n' \in \mathbb{Z}^+$  such that  $n' < n$ ;

$M(1) \leq M(1)$ ,  $M(2) \leq M(2)$ , and  $M(1) = c \leq 2c + 2d = M(2)$  since  $c, d \geq 0$ .

So  $R(1)$  and  $R(2)$  are true.

Consider  $n \geq 2$

Then  $1 \leq \lfloor \frac{n}{2} \rfloor \leq \lceil \frac{n}{2} \rceil \leq n - 1 < n$

So  $R(\lfloor \frac{n}{2} \rfloor)$ ,  $R(\lceil \frac{n}{2} \rceil)$ , and  $R(n - 1)$  all hold

Let  $m \in \mathbb{Z}^+$  be arbitrary and suppose  $m \leq n$ .

**Case 1.**  $m = n$ .

$M(m) = M(n)$  by substitution

**Case 2.**  $m = n - 1$ .

$$\begin{aligned} M(n-1) &= M\left(\left\lceil \frac{n-1}{2} \right\rceil\right) + M\left(\left\lfloor \frac{n-1}{2} \right\rfloor + d(n-1)\right) && \text{since } n-1 > 1 \\ &\leq M\left(\left\lfloor \frac{n}{2} \right\rfloor\right) && \text{since } \left\lceil \frac{n-1}{2} \right\rceil \leq \left\lfloor \frac{n}{2} \right\rfloor \leq n \\ &\quad + M\left(\left\lfloor \frac{n}{2} \right\rfloor\right) && \text{since } \left\lfloor \frac{n-1}{2} \right\rfloor \leq \left\lfloor \frac{n}{2} \right\rfloor \leq n \\ &\quad + dn && \text{since } d \geq 0 \\ &= M(n) \end{aligned}$$

**Case 3.**  $m < n - 1$ .

Then  $M(m) \leq M(n-1) \leq M(n)$  by induction hypothesis

In all cases  $M(m) \leq M(n)$

So  $(m \leq n) \text{ IMPLIES } (M(m) \leq M(n))$ .

Since  $m$  was arbitrary,  $R(n)$  holds.

By strong induction,  $\forall n \in \mathbb{Z}^+. R(n)$ .

QUOD  
ERAT  
DEM■

*Proof of Theorem 1.*

Let  $n \in \mathbb{Z}^+$  be arbitrary, assume  $n \geq 2$ .

Let  $2^i$  be the smallest power of 2 that is greater or equal to  $n$

Then,  $n \leq 2^i < 2n$  which implies  $i < \log_2(2n)$  and so

$$\begin{aligned} M(n) &\leq M(2^i) = c2^i + di2^i \\ &< 2cn + 2dn\log_2(2n) \\ &= 2cn + 2dn(1 + \log_2(n)) \\ &< (2c + 4d)n\log_2 n \end{aligned}$$

By generalization, we have  $\forall n \in \mathbb{N}. [(n \geq 2) \text{ IMPLIES } M(n) \leq (2c + 4d)n\log_2 n]$ , so  $M(n) \in O(n\log n)$ .

Let  $2^j$  be the largest power of 2 that is less than or equal to  $n$ .

Then, we have  $n \geq 2^j > \frac{n}{2}$  and so

$$\begin{aligned} M(n) &\geq M(2^j) \\ &= c2^j + dj2^j \\ &> c\frac{n}{2} + \frac{dn}{2}\log_2(\frac{n}{2}) \\ &\geq \frac{dn}{4}\log_2 n \quad \text{for } n \geq 4 \end{aligned}$$

Thus,  $M(n) \in \Omega(n\log n)$

Therefore, we conclude  $M(n) \in \Theta(n\log n)$ .

QUOD  
ERAT  
DEM■

## Analyzing the Worst Case Time Complexity of Iterative Algorithms

- Code without loops or procedure calls -  $O(1)$  step
- Loops: if a loop is executed  $O(f(n))$  times and each iteration takes  $O(g(n))$  steps, then the entire loop takes  $O(f(n) \cdot g(n))$  steps.
- If statements: Suppose  $A, B, C$  are blocks of code that take  $O(f(n)), O(g(n)), O(h(n))$  steps respectively. Then

---

```
if A
  B
else
  C
```

---

takes  $O(f(n) + \max\{g(n), h(n)\})$  steps.

- Procedure + function calls: Suppose that the worst case step complexity of a procedure (or function)  $\mathcal{P}$  with input of size  $r$  is  $T(r) \in O(f(r))$ , then a call to  $\mathcal{P}$  with an input of size  $g(n)$  takes  $O(f(g(n)))$  steps.

Consider the following Merge code:

```

function MERGE( $A[1..m], m, B[1..n], n$ )
     $i \leftarrow 1$ 
     $j \leftarrow 1$ 
     $h \leftarrow 1$ 

    while  $i \leq m$  and  $j \leq n$ 
        if  $A[i] < B[j]$ 
             $C[h] \leftarrow A[i]$ 
             $i \leftarrow i + 1$ 
        else
             $C[h] \leftarrow B[j]$ 
             $j \leftarrow j + 1$ 
         $h \leftarrow h + 1$ 

    if  $i > m$ 
        while  $j \leq n$ 
             $C[h] \leftarrow B[j]$ 
             $j \leftarrow j + 1$ 
             $h \leftarrow h + 1$ 
    else
        while  $i \leq m$ 
             $C[h] \leftarrow A[i]$ 
             $i \leftarrow i + 1$ 
             $h \leftarrow h + 1$ 

```

**Input size:**  $m, n$  (better than  $\max\{m, n\}, m + n$ )

**Step:** “increasement of  $i + j$ ” counting steps the exact same way as “assignment to  $C$  ( $m + n$ )”. We can also do “comparison between elements of  $A$  and elements of  $B$ ” (not as good as the previous 2).

**Time Complexity:**  $m + n - 1$ .

Step = “# of iterations of 1st while loop.”

1st while loop terminates as soon as  $i > m$  or  $j > n$ , so either  $i$  has been increased  $m$  times or  $j$  has been increased  $n$  times.

Exactly one of  $i$  and  $j$  is increased each iteration, so both conditions can't be true.

In the first case  $j$  is increased at most  $n - 1$  times.

In the second case  $i$  is increased at most  $m - 1$  times.

In both cases  $\leq m + n - 1$  comparisions are performed.

If  $T_{M \in RUR}(m, n)$  to the worst case number of comparisions between elements of  $A[1..m]$  and  $B[1..n]$  then  $T_{M \in RUR}(m, n) \leq m + n - 1$ .

*Why is  $m + n - 1$  a lower bound?*

For each  $m, n$  what are the examples of  $A[1..m]$  and  $B[1..n]$  for which  $n + m - 1$  comparisions are performed?

If all elements in  $A$  and  $B$  are less than  $A[m]$  and except for  $A[m]$  are less than  $B[n]$ .

$$A = \underbrace{[0 \cdots 0]}_n 2$$

$$B = \underbrace{[0 \cdots 0]}_n 1$$

$$T_{\text{Merge}}(m, n) \geq m + n - 1.$$

Let  $M : \mathbb{Z}^+ \rightarrow \mathbb{N}$  denote the worst case time complexity of MergeSort

$$M(n) = 0 \quad \text{if } n \leq 1$$

$$M(n) \leq M(\lceil \frac{n}{2} \rceil) + M(\lfloor \frac{n}{2} \rfloor) + n - 1 \quad \text{if } n > 1$$

---

```

function MERGESORT( $A[1..n]$ ,  $n$ ) ▷  $n = \text{input size}$ 
  if  $n \leq 1$ 
    return
   $m \leftarrow \lceil \frac{n}{2} \rceil$  ▷ step = comparision between elements of  $A$ 
   $A' \leftarrow A[1..m]$ 
   $A'' \leftarrow A[m+1..n]$ 
   $M \in \text{RURSORT}(A', m)$ 
   $M \in \text{RURSORT}(A'', n-m)$ 
   $A \leftarrow \text{Merge}(A[1..m], m, A[m+1..n], n-m)$ 

```

---

<sup>1</sup> Let  $M' : \mathbb{Z}^+ \rightarrow \mathbb{N}$  be  $M'(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ M'(\lceil \frac{n}{2} \rceil) + M'(\lfloor \frac{n}{2} \rfloor) + n - 1 & \text{if } n > 1 \end{cases}$

Then  $M'(n) \in O(n \log n)$  as proven at the start.

It is very easy to prove by induction  $\forall n \in \mathbb{Z}^+. M(n) \leq M'(n)$ .

Hence  $M(n) \in O(n \log n)$ .

---

<sup>1</sup>**Note:** We also need to show the linear bound is achievable