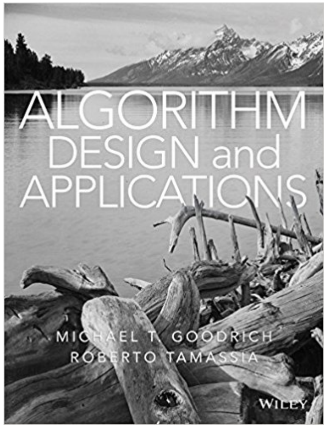
COMP26120: Introducing Complexity Analysis (2020/21)

Lucas Cordeiro

lucas.cordeiro@manchester.ac.uk

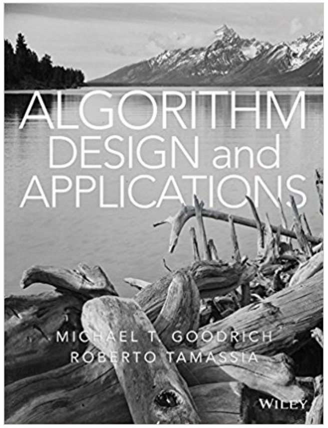
Introducing Complexity Analysis

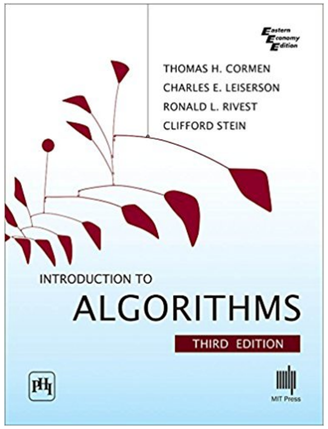
- Textbook:
  - *Algorithm Design and Applications,* Goodrich, Michael T. and Roberto Tamassia (Chapter 1)



Introducing Complexity Analysis

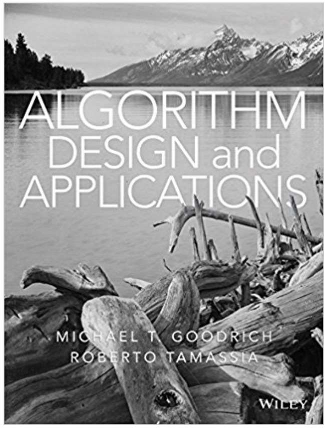
- Textbook:
  - *Algorithm Design and Applications,* Goodrich, Michael T. and Roberto Tamassia (Chapter 1)
  - *Introduction to Algorithms*, Cormen, Leiserson, Rivest, Stein (Chapters 2 and 3)

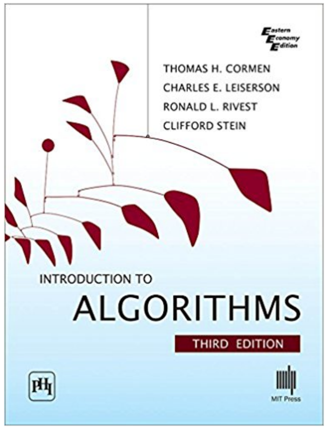


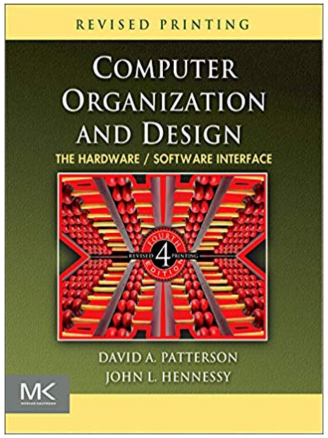


Introducing Complexity Analysis

- Textbook:
  - *Algorithm Design and Applications,* Goodrich, Michael T. and Roberto Tamassia (Chapter 1)
  - *Introduction to Algorithms*, Cormen, Leiserson, Rivest, Stein (Chapters 2 and 3)
  - *Computer Organization and Design,* Patterson and Hennessy (Chapter 2)






REVISED PRINTING
COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE / SOFTWARE INTERFACE
4
DAVID A. PATTERSON
JOHN L. HENNESSY
MK

Intended Learning Outcomes

- Define **asymptotic notation, functions,** and **running times**

Intended Learning Outcomes

- Define **asymptotic notation, functions,** and **running times**
- Analyse the **running time** used by an algorithm via **asymptotic analysis**

Intended Learning Outcomes

- Define **asymptotic notation, functions,** and **running times**
- Analyse the **running time** used by an algorithm via **asymptotic analysis**
- Examine and sketch examples of **asymptotic analysis** using the **insertion sorting algorithm**

Motivating Example

- What does this code fragment represent?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
    ●   t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

Motivating Example

- What does this code fragment represent?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
    ●   a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

Motivating Example

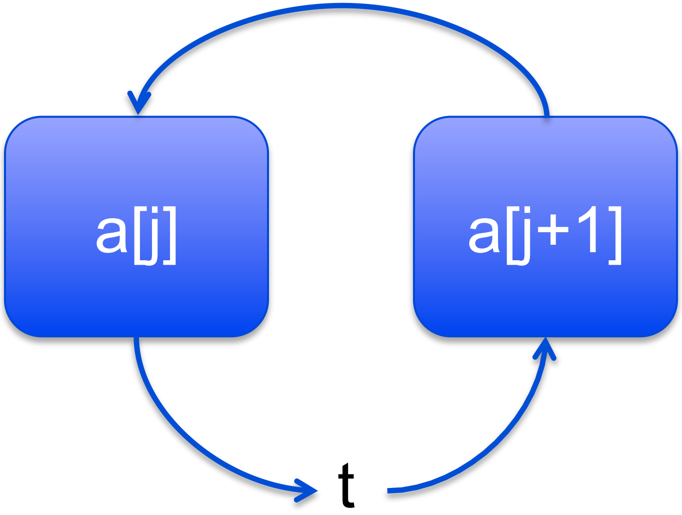
- What does this code fragment represent?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
    ●   a[j+1] = t;
      }
    }
  }
…
```

Motivating Example

- What does this code fragment represent?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```



Motivating Example

- What does this code fragment represent?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

- This is bubble sort

Motivating Example

- What does this code fragment represent? What is its complexity?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

- This is bubble sort

Motivating Example

- What does this code fragment represent? What is its complexity?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

- This is bubble sort
- There are two loops

Motivating Example

- What does this code fragment represent? What is its complexity?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

  - This is bubble sort
  - There are two loops
  - Both loops make _n-1_ iterations so we have _(n-1)*(n-1)_

Motivating Example

- What does this code fragment represent? What is its complexity?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

- This is bubble sort
- There are two loops
- Both loops make $n-1$ iterations so we have $(n-1)*(n-1)$
- The complexity is $O(n^2)$

Motivating Example

- What does this code fragment represent? What is its complexity?

```
…
for(i = 0; i < N−1; i++) {
    for(j = 0; j < N−1; j++) {
      if (a[j]> a[j+1]) {
        t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
…
```

- This is bubble sort
- There are two loops
- Both loops make *n-1* iterations so we have *(n-1)*(n-1)*
- The complexity is $O(n^2)$

Perform worst case analysis and ignore constants

Motivating Example

You can download these examples from the course website:

http://syllabus.cs.manchester.ac.uk/ugt/2020/COMP26120/