

COL106: Data Structures and Algorithms

Ragesh Jaiswal, IITD

Administrative Information

Administrative Information

- Course Instructor:
 - Ragesh Jaiswal
 - *Email:* rjaiswal@cse.iitd.ac.in
 - Office: SIT 403
- Course Time/Place:
 - Lectures:
 - Tue, Thu, Fri 11-12, LH 111
- Teaching Assistants:
 - TBD

Administrative Information

- Grading Scheme
 - ① *Homework (programming)*: 20%
 - ② *Quizzes*: 20% (weekly)
 - ③ *Minor*: 30% (two minors 15% each)
 - ④ *Major*: 30%
- Homework and Quizzes:
 - Schedule for completion date of homework is posted on the course page. **Homework will not be graded.**
 - **However**, there will be a quiz every week on the material given in the homework for the past week.
 - Gradescope: A paperless grading system. Use the course code **9Z547M** to register. **Please use your formal email address from IIT Delhi.**
- Policy on cheating: **Students using unfair means will be severely penalised.**

- Textbooks: *I will be following this book very closely. So, it will be a good idea to get a copy of this book.*
 - ① Data Structures and Algorithms in Java by *Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser.*
- Other reference books:
 - Introduction to Algorithms by *Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein.*
- Course webpage:
<http://www.cse.iitd.ac.in/~rjaiswal/2017/COL106/>
 - The site will contain course information, references, homework, course slides etc. Please check this page regularly.

Administrative Information

- Programming language: *Java*
- You are expected to be comfortable with the first two chapters of the Textbook. These chapters are introductions to Basic Java Programming and Object Oriented Programming.
- There is a **java module** on the course page. You are expected to go through this on your own.
- There will be a **quiz** based on the java module (*first three sessions only*).

Data Structures and Algorithms

- Data Structure: Systematic way of organising and accessing data.
- Algorithm: A step-by-step procedure for performing some task.

- How do we describe an algorithm?
 - Algorithms are platform independent and so should be their description.
 - This allows us to focus on the main ideas rather than spend time parsing the programming language specific syntax and the implementation details.

- How do we describe an algorithm?
 - Algorithms are platform independent and so should be their description.
 - This allows us to focus on the main ideas rather than spend time parsing the programming language specific syntax and the implementation details.
 - A concise way of describing algorithm is **pseudocode**.
 - Pseudocode is not an actual code.
 - It consists of:
 - high-level programming constructs (if-then, for etc.) +
 - natural language.

Introduction

- How do we describe an algorithm?
 - Algorithms are platform independent and so should be their description.
 - This allows us to focus on the main ideas rather than spend time parsing the programming language specific syntax and the implementation details.
 - A concise way of describing algorithm is **pseudocode**.
 - Pseudocode is not an actual code.
 - It consists of:
 - high-level programming constructs (if-then, for etc.) + natural language.

Algorithm

```
FindMin( $A, n$ )  
-  $min \leftarrow A[1]$   
- for  $i = 2$  to  $n$   
  - if ( $A[i] < min$ )  
    -  $min \leftarrow A[i]$   
- return( $min$ )
```

- How do we describe an algorithm?
 - Algorithms are platform independent and so should be their description.
 - This allows us to focus on the main ideas rather than spend time parsing the programming language specific syntax and the implementation details.
 - A concise way of describing algorithm is **pseudocode**.
 - Pseudocode is not an actual code.
 - It consists of:
high-level programming constructs (if-then, for etc.) + natural language.

Algorithm

```
FindMin( $A, n$ )  
-  $min \leftarrow A[1]$   
- for  $i = 2$  to  $n$   
  - if  $A[i]$  is smaller than  $min$   
    -  $min \leftarrow A[i]$   
- return( $min$ )
```

- How do we describe an algorithm?
 - Using a **pseudocode**.
- What are the desirable features of an algorithm?

- How do we describe an algorithm?
 - Using a **pseudocode**.
- What are the desirable features of an algorithm?
 - It should be correct.
 - It should run fast.
 - It should take small amount of space (RAM).
 - It should consume small amount of power.
 - ⋮

- How do we describe an algorithm?
 - Using a **pseudocode**.
- What are the desirable features of an algorithm?
 - 1 **It should be correct.**
 - 2 It should run fast.
- How do we argue that an algorithm is correct?

End