COMP 2011: Data Structures

Lecture 1. Introduction

Dr. CAO Yixin

yixin.cao@polyu.edu.hk

September 09, 2020

Tutors

- Ke, Yuping, yuping.ke@connect.polyu.hk
- Wang, Shenghua, shenghua.wang@connect.polyu.hk
- Zhou, Qihua, qi-hua.zhou@connect.polyu.hk

Please only send personal inquiries to the instructor and tutors, e.g., grading issues.
It's better to post questions of general interest on the discussion forum.
You'll get quicker responses there, and have chance to win bonus points.

Assessment

- Assignments & Labs: 20%
- Quizzes: 20%
- Mid-term exam: 20%
- Final exam: 40%

Everything on Blackboard
slides; videos, reading materials; assignment and answers; etc.

Assessment

- Assignments & Labs: 20%
- Quizzes: 20%
- Mid-term exam: 20%
- Final exam: 40%

- make your code as simple as possible;
- make sure that your code compiles; and
- format you code.

  (Assignment 1 will explain the homework submission process in detail.)

- No late submission will be entertained.

Everything on Blackboard

slides; videos, reading materials; assignment and answers; etc.

Quiz arrangements

Each quiz will have a large number of questions, don't try to answer all of them. I myself cannot do it in the quiz time. The purpose is to keep you busy so that you don't have time to "help" others.

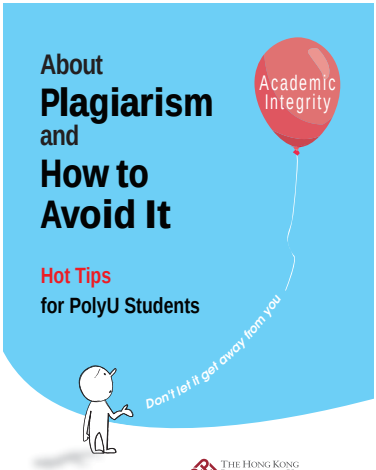
For each quiz, there will be a grading number n (not announced before the quiz). Suppose that you've c correct answers and w wrong answers, your score of the quiz is

$$\begin{cases} \frac{c}{c+w} * 100 & \text{if } c+w \geq n \\ \frac{c}{n} * 100 & \text{otherwise,} \end{cases}$$

and if $c > n$, then you get bonus $(c-n) * 0.5$, disregarding the number w.
For example, if there are 50 questions and $n = 20$, then the score is

- 90 for $c = 40$ and $w = 10$;
- 80 for $c = 20$ and $w = 5$;
- 75 for $c = 15$ and $w = 5$;
- 50 for $c = 10$ and $w = 5$;
- 50 for $c = 10$ and $w = 0$;

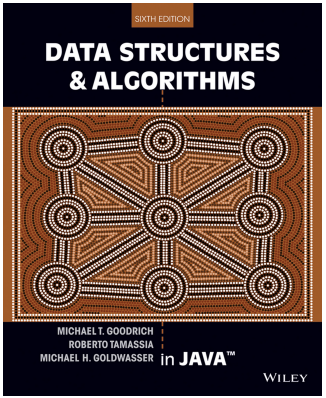
- Don't read other's codes.
- Don't share your codes.

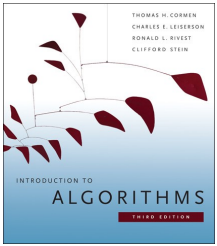


http://edc.polyu.edu.hk/PSP/Plagiarism_Booklet.pdf

A hard copy can be found at the GO (PQ806).

Textbooks





Too difficult for 2011,
but important for a deep understanding.


Hola
こんにちは
Guten Tag
Привет
Hello
Ciao
Olá
Bonjour
مرحباً
你好

```
int recFind(int key, int low, int high) {
    int mid = (low + high) / 2;
    if (a[mid] == key) return mid;
    if (low == high) return -1;
    if (a[mid] < key) return recFind(key, mid, high);
    return recFind(key, low, mid);
}

int recFind(int key) {
    if (length == 0) return -1;
    return recFind(key, 0, length - 1);
}
```

Can somebody tell the language of this snippet?

For our purpose, it doesn't matter which programming language is used.

```
int recFind(int key, int low, int high) {
    int mid = (low + high) / 2;
    if (a[mid] == key) return mid;
    if (low == high) return -1;
    if (a[mid] < key) return recFind(key, mid, high);
    return recFind(key, low, mid);
}

int recFind(int key) {
    if (length == 0) return -1;
    return recFind(key, 0, length - 1);
}
```

Can somebody tell the language of this snippet?

For our purpose, it doesn't matter which programming language is used.

Java

Arrays in Java

Java provides three different ways to define an array:

- `int[] a; a = new int[7];`
- `int[] a = new int[7];`
- `int[] a = {11, 16, 9, 4, 12, 30, 3};`

Arrays in Java

Java provides three different ways to define an array:

- `int[] a; a = new int[7];`
- `int[] a = new int[7];`
- `int[] a = {11, 16, 9, 4, 12, 30, 3};`

Arrays in Java

Java provides three different ways to define an array:

- `int[] a; a = new int[7];`
- `int[] a = new int[7];`
- `int[] a = {11, 16, 9, 4, 12, 30, 3};`

Arrays in Java

Java provides three different ways to define an array:

- `int[] a; a = new int[7];`
- `int[] a = new int[7];`
- `int[] a = {11, 16, 9, 4, 12, 30, 3};`

what if `x = a[2];`

Arrays in Java

Java provides three different ways to define an array:

- `int[] a; a = new int[7];`
- `int[] a = new int[7];`
- `int[] a = {11, 16, 9, 4, 12, 30, 3};`

what if `a[5] = 5;`

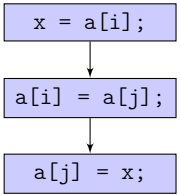
Arrays in Java

Java provides three different ways to define an array:

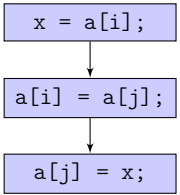
- `int[] a; a = new int[7];`
- `int[] a = new int[7];`
- `int[] a = {11, 16, 9, 4, 12, 30, 3};`

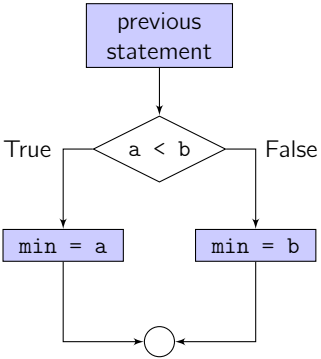
$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
11	16	9	4	12	30	3

Basic control flow



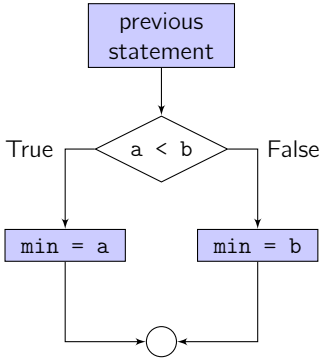
Basic control flow





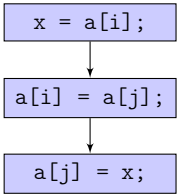
Basic control flow

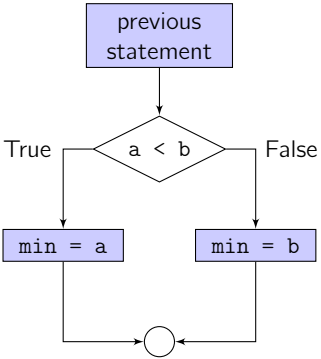
```
x = a[i];
a[i] = a[j];
a[j] = x;
```

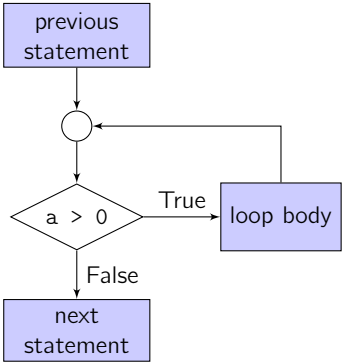



```
min = a < b? a : b;
```

Basic control flow

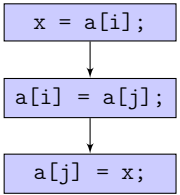


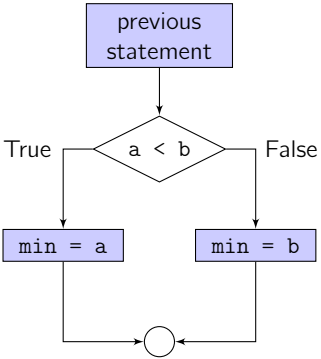


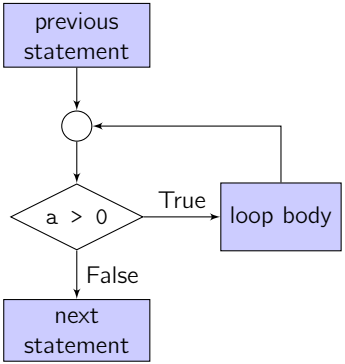



```
min = a < b? a : b;
```

Basic control flow








```
min = a < b? a : b;
```

```
while (a > 0) {
    ...
    a--;
}
```

Which one is wrong?

```
while (true) {
    ...
}
```

```
for (i = 1; i = 10; i++) {
    ...
}
```

Which one is wrong?

```
while (true) {
    ...
}
```

```
for (i = 0.1; i != 1; i+=0.1) {
    ...
}
```

Control flow II: subroutine call

return value type — **function name** — **list of formal parameters**

```
int puissance (int x, int n) {

    int i, p = 1;

    for (i = 1; i <= n; i++)
        p = p * x;

    return p;
}
```

local variables declaration

instructions

instruction `return`

Java has provided all the basic structures, as well as all the algorithms we'll discuss during this course.

My implementations, mostly stupid, are surely worse than those from Java; they are for demonstration purpose only.

Java has provided all the basic structures, as well as all the algorithms we'll discuss during this course.

My implementations, mostly stupid, are surely worse than those from Java; they are for demonstration purpose only.

Important links

(Almost) everything about java can be found at:
Java SE 11 API Specifications (for download)

Java collections (data structures) documents:

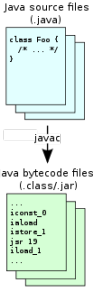
- Collections Framework
- Collections Framework Tutorial

Java 14 is no longer free, so please don't use it unless you've a good reason.

How Java works (the very basic)

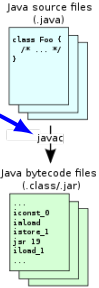
How Java works (the very basic)

A Java source file (`.java`) is compiled (using `javac`), which generates the Java bytecode file (`.class`).



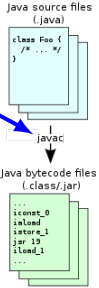
How Java works (the very basic)

A Java source file (`.java`) is compiled (using `javac`), which generates the Java bytecode file (`.class`).



How Java works (the very basic)

A Java source file (`.java`) is compiled (using javac), which generates the Java bytecode file (`.class`).

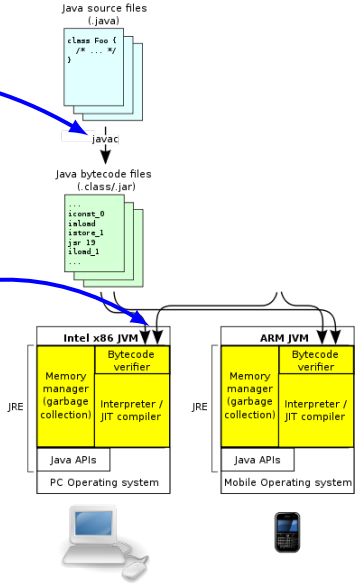


A Java bytecode file (`.class`) can be executed on any JVM, using java.

How Java works (the very basic)

A Java source file (`.java`) is compiled (using `javac`), which generates the Java bytecode file (`.class`).

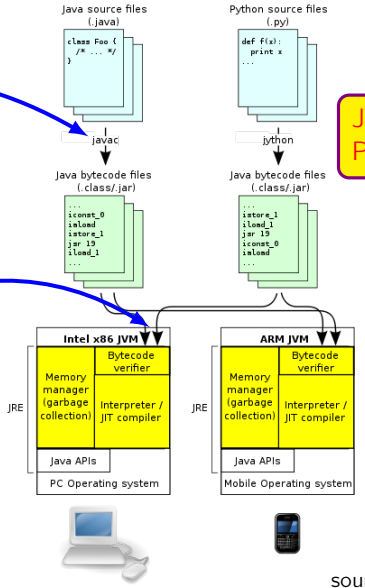
A Java bytecode file (`.class`) can be executed on any JVM, using `java`.



How Java works (the very basic)

A Java source file (`.java`) is compiled (using `javac`), which generates the Java bytecode file (`.class`).

A Java bytecode file (`.class`) can be executed on any JVM, using `java`.



source: stackoverflow

Basics of Algorithms

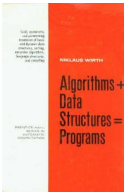
Why study data structures + algorithms?

I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships.

Linus Travolds

Algorithms + Data Structures = Programs.

Niklaus Wirth



Data structures are

ways to organize data (information).
examples:

- simple variables—primitive types
- objects — collection of data items of various types
- arrays — collection of data items of the same type, stored contiguously
- linked lists — sequence of data items, each one pointing to the next one
- stack, queue, tree, table, graph...

An algorithm is (like) a recipe

Input → Procedure → Output

Ingredients

- $1\frac{1}{2}$ cups all-purpose flour
- $3\frac{1}{2}$ teaspoons baking powder
- 1 teaspoon salt
- 1 tablespoon white sugar
- $1\frac{1}{4}$ cups milk
- 1 egg
- 3 tablespoons butter, melted

Directions

1. In a large bowl, sift together the flour, baking powder, salt and sugar.
   Make a well in the center and pour in the milk, egg and melted butter; mix until smooth.
2. Heat a frying pan over medium high heat.
   Pour or scoop the batter onto the pan, using approximately $\frac{1}{4}$ cup for each pancake.
   Brown on both sides and serve hot.

Source: allrecipes.com (demo)

Problem solving with a computer

1. model the problem as a computational problem,
2. design an algorithm,
3. implement the algorithm as a program.

Problem solving with a computer

1. model the problem as a computational problem,
2. design an algorithm,
3. implement the algorithm as a program.

you need to choose data structures

Problem solving with a computer

1. model the problem as a computational problem,
2. design an algorithm,
3. implement the algorithm as a program.

you need to choose data structures

you may have to revise data structures

Problem solving with a computer

1. model the problem as a computational problem,
2. design an algorithm,
3. implement the algorithm as a program.

you need to choose data structures

you may have to revise data structures

source: woodsideeggs.com

Expressing algorithms

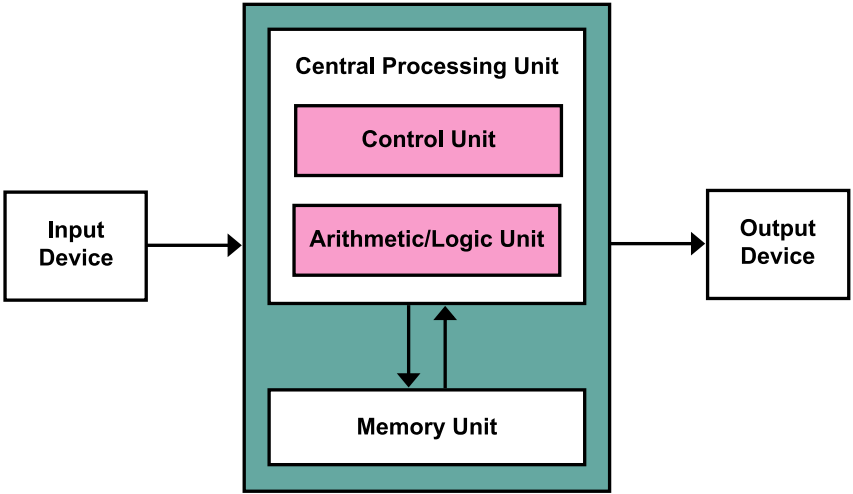
- English description;
- a real programming language;
- pseudocode (more precise than English, less boring than programming languages)

A common mistake my students make is to use pseudo-code to dress up an ill-defined idea so that it looks more formal.

Skiena, *The Algorithm Design Manual,*

1. heat a pan
2. add oil
3. add water
4. ...

Essentials of Computer Science



https://en.wikipedia.org/wiki/Von_Neumann_architecture

Which one runs faster?

```
for (i = 0; i < N; i++)
  for (j = 0; j < N; j++)
    sum += a[i][j];
```

```
for (j = 0; j < N; j++)
  for (i = 0; i < N; i++)
    sum += a[i][j];
```

In old versions of Java, one is thousand times faster than the other (N=16384).

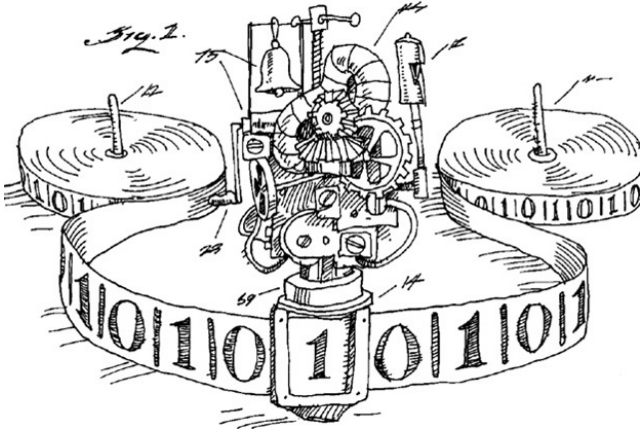
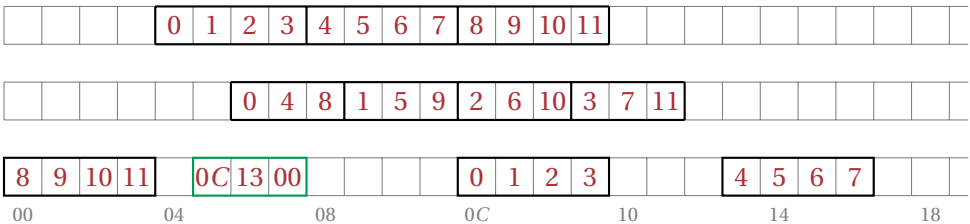


Fig. 1.
1 0 1 1 0 1 0 1 1 0 1
1 0 1 1 0 1 0 1 1 0 1 1 0

Two-dimensional arrays

Now that the memory is a "tape," how to accommodate a two-dimensional array?

Three ways to store $\begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \end{bmatrix}$



Our model

We abstract all the details,
left with one core, one memory (one-dimensional array).

George Box:
All models are wrong but some are useful.

Appetizer

Which one is better?

source: Amazon

Which one is better?

source: Amazon

Which one is better?

source: Amazon

Message 1:
How your data are organized
depends on
your applications.

Physical addresses

Why street numbers are usually non-consecutive?

Physical addresses

Why street numbers are usually non-consecutive?

Hong Kong Government:

There should be a reasonable balance
between the odd numbers on one side and
the even numbers on the other side of a street.

Physical addresses

Why street numbers are usually non-consecutive?

Hong Kong Government:

There should be a reasonable balance between the odd numbers on one side and the even numbers on the other side of a street.

Message 2:
Be prepared for change.

Food Court

Food Court

Message 3:
Data structures are everywhere.

Data structures are easy:
All materials are (naively) natural. They are discovered instead of invented.

Data structures are difficult:
Knowing the concepts is not enough; You have to practice to master them.

Course topics

- vocabulary for design and analysis of algorithms
  - Dry and boring, but that is what defines computing, and defines us.
  - Everybody in PolyU can program; everybody in FENG knonws machine learning.
  - The difference between CS and others lie in data structures and algorithms.
- use and implementation of data structures
- design paradigm (divide and conquer)
- how to sort and how to not sort.

Skills you'll learn

- improve your programming
- sharpen your mathematical and analytical skills
- start "thinking algorithmically"
- Prepare your technical interviews.

Basic sorting algorithms

Three basic sorting algorithms

- bubble sort
- selection sort
- insertion sort

Do you still remember how to write them?

Which one is the best, the worst?

```
void bubble(int[] a) {
  int n = a.length;
  int i, j;
  for (i=1; i<n; i++)
    for (j=0; j<n-i; j++)
      if (a[j+1] < a[j])
        swap(a, j, j+1);
}
```

```
void insertion(int[] a) {
  int i, j, key, n = a.length;
  for (i = 1; i < n; i++) {
    key = a[i];
    for (j = i - 1; j >= 0; j--) {
      if (a[j] <= key) break;
      a[j + 1] = a[j];
    }
    a[j + 1] = key;
  }
}
```

```
void selection(int[] a) {
  int n = a.length;
  int min;
  for (int i=0; i<n-1; i++) {
    min = i;
    for (int j=i+1; j<n; j++)
      if (a[min] > a[j]) min = j;
    swap(a, min, i);
  }
}
```

I compress the codes to fit into one slide,
please do not do that in your programming.

The first algorithms you've learned

We're taught that multiplications are more difficult than additions.

$$\begin{array}{r} 2011 \\ +\ 1234 \\ \hline 3245 \end{array}$$

4 additions

$$\begin{array}{r} 2011 \\ \times\ 1234 \\ \hline 8044 \\ 6033\cdot \\ 4022\cdot\cdot \\ 2011\cdot\cdot\cdot \\ \hline 2481574 \end{array}$$

4×4 additions

$1+2+3+2+1$ additions

The first algorithms you've learned

We're taught that multiplications are more difficult than additions.

$$\begin{array}{r} 2\,0\,1\,1 \\ +\;\; 1\,2\,3\,4 \\ \hline 3\,2\,4\,5 \end{array}$$

4 additions

$$\begin{array}{r} 2\,0\,1\,1 \\ \times\;\; 1\,2\,3\,4 \\ \hline 8\,0\,4\,4 \\ 6\,0\,3\,3\,\cdot \\ 4\,0\,2\,2\,\cdot\,\cdot \\ 2\,0\,1\,1\,\cdot\,\cdot\,\cdot \\ \hline 2\,4\,8\,1\,5\,7\,4 \end{array}$$

4×4 additions

$1+2(+1)+3+2+1$ additions because there is a carry

A formal treatment

- A smart student would calculate 1234×2011 instead of 2011×1234
- A simple change of the order of operands save near half work.
- Even faster if the second operand is 1000.
- But which is faster: 6789×9876 or 9876×6789?

- We've seen one carry.
- There can be up to four carries in the addition, even more for the multiplication.
- Our belief is not shaken even 1000×1000 is obviously simpler than $6789 + 9876$.

The belief that multiplications are more difficult than additions hasn't be proved.

Week 2: Analysis of Algorithms