

CPE112RC

Module1-3

Course Orientation

# Course Orientation

Dr. Prapong Prechaprapranwong



# CPE112:

## Programming with Data Structures

---

### Learning Outcome:

1. Explain properties and applications of linear data structures
2. Explain properties and applications of non-linear data structures
3. Select and implement appropriate data structures and associated algorithms to efficiently solve programming problems

Programming based on



python

Try to do your codes better than you have done in CPE100

# Native Data Structures in Python

There are 4 native data structures in the Python:

- **List** - ordered and changeable
  - Allows duplicate members
- **Tuple** - ordered and unchangeable
  - Allows duplicate members
- **Set** - unordered, unchangeable (remove or add only), and unindexed
  - No duplicate members
- **Dictionary** - ordered (from Python 3.7) and changeable
  - No duplicate members



## Abstract Data Type (ADT)

definition for a data type

- a set of values
- a set of operations allow on data type

# The stack abstract data type

an ordered collection of items where items are added to and removed from the end called the “top” (LIFO).

The stack operations are given below.

- <sup>สร้างกอง</sup> **Stack()** creates a new stack that is empty. It needs no parameters and returns an empty stack.
- <sup>เอาค่าเข้าไว้</sup> **push(item)** adds a new item to the top of the stack. It needs the item and returns nothing.
- <sup>เอาด้านบนสุดออก</sup> **pop()** removes the top item from the stack. It needs no parameters and returns the item. The stack is modified.
- <sup>ดูด้านบนสุด</sup> **peek()** returns the top item from the stack but does not remove it. It needs no parameters. The stack is not modified.
- <sup>ตรวจสอบว่าว่างเปล่าหรือไม่</sup> **isEmpty()** tests to see whether the stack is empty. It needs no parameters and returns a Boolean value.
- <sup>นับว่ากองมีกี่ชิ้น</sup> **size()** returns the number of items on the stack. It needs no parameters and returns an integer.

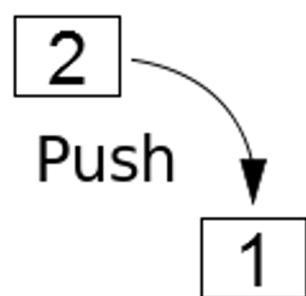
PUSH



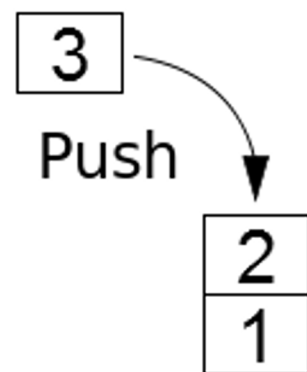
POP



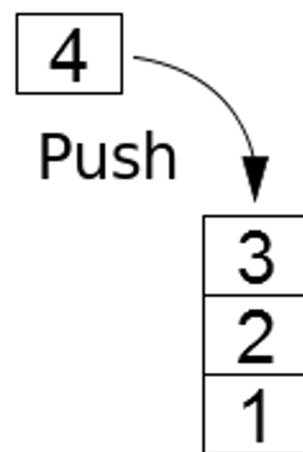
1



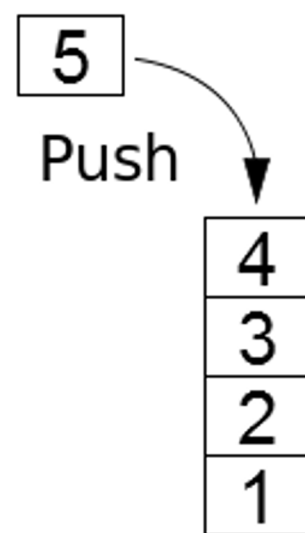
2



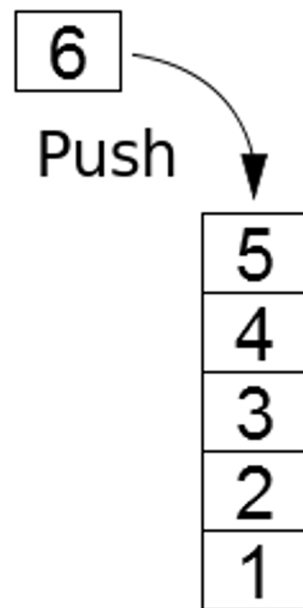
3



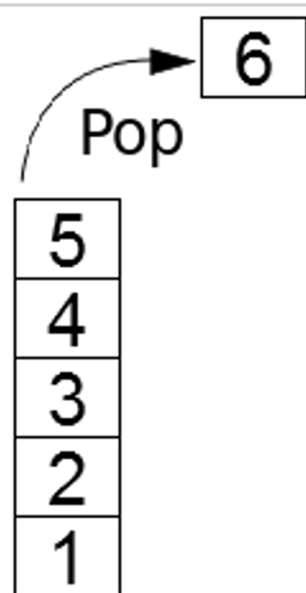
4



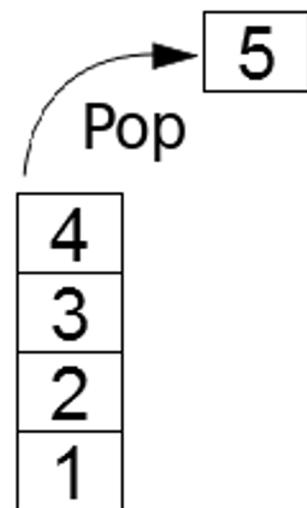
5



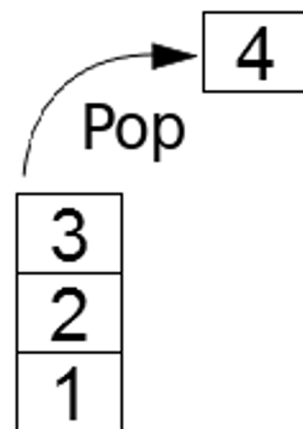
6



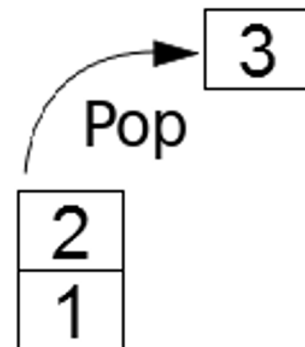
7



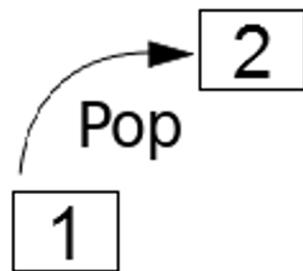
8



9

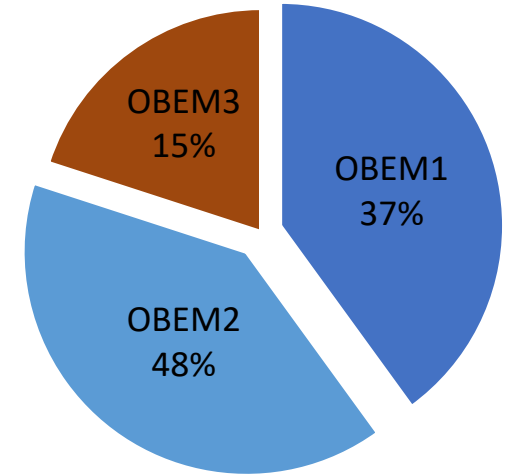


10



# OBEM

- OBEM1: Linear Data Structure
- OBEM2: Non-Linear Data Structure
- OBEM3: Implementation of Data Structure in Machine Learning





# Tools for Learning and Lab Assignment



- แต่ละ สัปดาห์จะมี Lab assignment ให้ทำเป็นกลุ่ม
- โดยให้เขียนสรุปความเข้าใจที่เรียนเป็นรายงาน บันทึกการทดลอง และตอบคำถามด้วย

<https://jupyter.org/>

# Divide and Conquer

## หลักการของ Divide and Conquer

เป็นวิธีการแก้ปัญหาที่ซับซ้อนโดยการนำปัญหามาแยกเป็นปัญหาย่อยๆ จากนั้นนำผลของปัญหาย่อยมาประกอบกันเพื่อหาคำตอบของปัญหาต้นฉบับ

- 1. Divide แบ่งปัญหามาทำเป็นปัญหาย่อยๆ โดยพิจารณาจากเงื่อนไขที่กำหนด
- 2. Conquer แก้ปัญหาย่อยๆ ที่ได้ออกมา
- 3. Combine นำปัญหาย่อยๆ ที่ได้ออกมาประกอบกันเพื่อหาคำตอบของปัญหาต้นฉบับ

Note: เราสามารถนำปัญหามาทำเป็นปัญหาย่อยๆ ได้โดยที่ไม่จำเป็นต้องใช้การแบ่งส่วนข้อมูลเสมอไป

Note 2: Divide and Conquer จะทำงานโดยลดขนาดของปัญหาลงเรื่อยๆ โดยที่ขนาดของปัญหาย่อยๆ จะลดลงด้วยอัตราคงที่ (Constant factor) หรือจะลดลงด้วยอัตราที่ลดลงเรื่อยๆ ก็ได้

## Algorithm ที่นำหลักการของ Divide and Conquer มาใช้

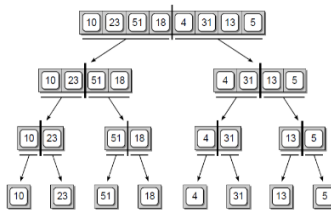
- 1. Merge Sort
- 2. Quick Sort
- 3. Multiplication of Large Integers
- 4. Strassen's Matrix Multiplication
- 5. Closest-Pair Problem by Divide and Conquer
- 6. Convex-Hull Problem by Divide and Conquer

### 1. Merge Sort

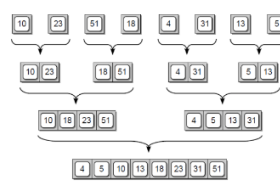
เป็นวิธีการเรียงลำดับที่นำหลักการของ Divide and Conquer มาใช้ โดยแบ่งข้อมูลออกเป็น 2 ส่วนๆ แล้วนำผลของปัญหาย่อยมาประกอบกันเพื่อหาคำตอบของปัญหาต้นฉบับ

#### ขั้นตอนการทำงานของ Merge Sort

1. แบ่งข้อมูลออกเป็นปัญหาย่อยๆ จนถึงขั้นที่เล็กที่สุด



2. นำคำตอบจากปัญหาย่อยๆ มาประกอบกันเพื่อหาคำตอบของปัญหาต้นฉบับ





Select Kernel

Python

Python

## 2. Quick Sort

ตัวอย่างที่ 2: Algorithm สำหรับการแบ่งส่วนที่เรานำมาใช้กับ Quick Sort

- ### 2.1) Lomuto Partition

### ขั้นตอนการทำ Lomuto Partition

- pivot -> เป็นค่าหลักที่จะนำมาใช้ในการเรียงเพื่อแบ่งได้ และจะเป็นค่าที่ต้นระหว่างมีซ้ายและมีขวาในคอนทาส
- r -> เป็นค่าที่จะระบุตำแหน่งตัวสุดท้ายของค่าในฝั่งซ้าย
- i -> ใช้ในการระบุค่าตัวถัดไปจาก pivot

- ### Practice 1: ប្រែប្រួល Quick Sort តាមវិធី Lomuto Partition

*P. S.*

ให้นักศึกษาแบ่งกลุ่มละ 3 คน และ ลงชื่อใน google form:

<https://forms.gle/BXcjg3YyhQb6ELG6>





LEB2.kmutt.ac.th

ส่ง Assignment บน LEB2



และ ส่งก่อน 23:59

# Quiz

ในวิชา มีการ Quiz ทั้งหมด 5 ครั้ง ให้นักศึกษาทำด้วยตนเอง

- ให้นักศึกษาตอบคำถามแบบปรนัย ประมาณ 5-10 ข้อ
- เมื่อทำเสร็จจะทราบคะแนนทันที
- ให้นักศึกษาสามารถ ทำ Quiz ได้ 3 ครั้ง

# Practice Examination

ในวิชา มีการสอบปฏิบัติ 2 ครั้ง

- ให้นักศึกษาเขียนโปรแกรมตามที่โจทย์กำหนดด้วยตนเอง
- โดยใช้ module (class) ที่พัฒนาขึ้นเอง
- ให้นักศึกษาทำให้ทันเวลาที่กำหนดแล้ว ส่งทันที
- ในกรณีที่ทำไมทัน ส่งเข้าได้ไม่เกิน 1 วัน แต่คะแนนเต็มจะถูกหักไป 30%
- ในกรณีที่พบว่า มีการลอกโค้ดเพื่อนำมาส่ง จะได้รับ 0 ทั้งผู้ลอกและต้นฉบับ

# Reference books

- R0.1 Full speed python
- R0.2 Python Tutorial
- R1 Algorithms in Python
- R2 Data Structures and Algorithms in Python
- R3 Data Structures and Algorithms Using Python
- R4 Open Data Structures (in pseudocodes)