

Project Array Part II

COS2103

Data structure and Algorithm

1 / 2568

Present to

Asst.Prof.DR.Chouvalit Khancome

By Krittaya Tantichaiyakul

6705500269

หัวข้อ

อุปกรณ์กีฬา



Assignment

COS2103 เทอม 1/68

☰ K

งาน 1.2 ArrayLists

Asst.Prof.Dr.Chouvalit Khancome • 5 ส.ค. (แก้ไข 16 ด.ค.)
50/50 ครบกำหนด 18 ด.ค.

ให้พัฒนา แอปพลิเคชันที่ออกแบบเหมือนกับ งาน 1.1 แต่ใช้ java.util.ArrayList แทนโครงสร้างข้อมูลที่เรียนรู้ ของอาจารย์ และใช้วิธีการอธิบายได้อย่างอิสระ สังเคราะห์ในสักษณะเดียวกัน

8 ความคิดเห็นในขั้นเรียน
เพิ่มความคิดเห็น

งานบอกรายงาน ให้คะแนนแล้ว

ProjectArrayPar... Compressed Archive X

การทำงานของโปรเจกต์ PDF X

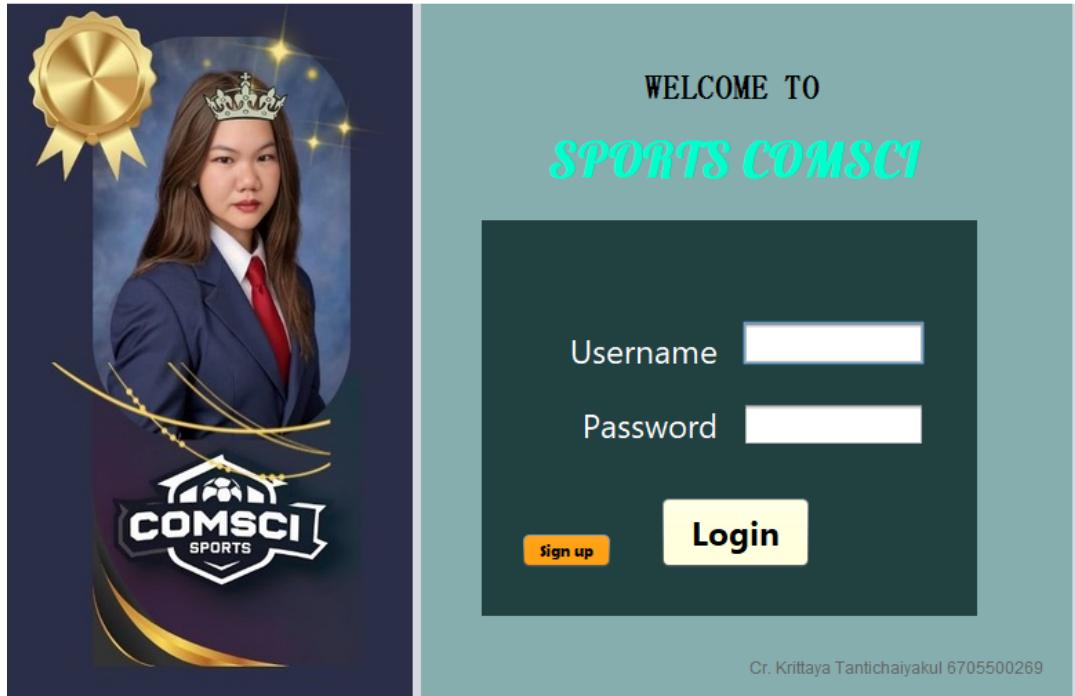
+ เพิ่มหรือสร้าง

ส่งรีบครั้ง

ความคิดเห็นล่าสุด
เพิ่มความคิดเห็นถึง Asst.Prof.Dr.Chouvalit Khancome

Program Interface Preview

1. Login page - Login



2. Category of array - Menu

The menu interface has a dark theme. At the top left is the "COMSCI SPORTS" logo. The center features the title "DATA STRUCTOR" above a "Project Array" section. On the right, there's a circular profile picture of a person and the text "Host : Krittaya001". The left sidebar contains five buttons: "Sneaker", "Soccer", "Golf", "Racket", and "Ball". The main content area shows a product detail for a "Adidas sneaker" with code 10233, type 1, and price 4600.0 baht. Navigation buttons "Back" and "Next" are at the bottom of this section. To the right, there's a "Manage data" button and a "Stock" section showing storage (-15), amount (13), and list no. (1). At the bottom, there are navigation keys [K][R][I][T][T][A] and a footer with the name "Krittaya Tantichaiyakul 6705500269".

3. list of products – Manage

No.	Code	Name	Price	Type	Address
1	10233	Adidas sne...	4600.0	1	/sc/sneake...
2	10472	Nike sneaker	4200.0	1	/sc/sneake...
3	10785	Skecher sn...	2500.0	1	/sc/ske.jpg
4	20096	Umbro Glove	800.0	2	/sc/ub.jpg
5	20504	Nike footba...	1500.0	2	/sc/shirtNik...
6	30319	Ping stick g...	1300.0	3	/sc/golfstic...
7	30873	Anyday golf ...	5000.0	3	/sc/golfbag....
8	40152	Yonex strike	2400.0	4	/sc/yon.png
9	40197	Kawasaki s...	1700.0	4	/sc/kawa.jpg
10	40601	Victor shuttle...	200.0	4	/sc/shuttle.j...
11	50290	Puma footb...	800.0	5	/sc/puma.jpg
12	50475	Mikasa voll...	1200.0	5	/sc/volley.jpg
13	50778	Wilson bas...	1500.0	5	/sc/wilson.j...

4. ไฟล์คลาสจัดเก็บข้อมูล – DataStore

```

11  public class DataStore {
12      public static int n=13;
13      final int UB=15;
14      int LB=0;
15      public static class Products {
16          public int code;
17          public String name;
18          public int type;
19          public float price;
20          public String pic;
21
22          public void setAll(int a, String b, int c, float d, String e){
23              code = a;
24              name = b;
25              type = c;
26              price = d;
27              pic = e;
28          }
29          static int getn(){
30              return n;
31          }
32      }
33  }

35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

```

```

public static Products[] list = new Products[15];
static {
    for (int i = 0; i < list.length; i++) {
        list[i] = new Products();
    }
}

// กำหนดชื่อเมืองต่างๆ
list[0].setAll(10233,"Adidas sneaker",1,4600,"/sc/sneakerAdi.jpg");
list[1].setAll(10472,"Nike sneaker",1,4200,"/sc/sneakerNike.jpg");
list[2].setAll(10785,"Skecher sneaker",1,2500,"/sc/ske.jpg");
list[3].setAll(20096,"Umbro Glove",2,800,"/sc/ub.jpg");
list[4].setAll(20504,"Nike football shirt",2,1500,"/sc/shirtNike.jpg");
list[5].setAll(30319,"Ping stick golf",3,1300,"/sc/golfstick.jpg");
list[6].setAll(30873,"Anyday golf bag",3,5000,"/sc/golfbag.jpg");
list[7].setAll(40152,"Yonex strike",4,2400,"/sc/yon.png");
list[8].setAll(40197,"Kawasaki strike",4,1700,"/sc/kawa.jpg");
list[9].setAll(40601,"Victor shuttlecock",4,200,"/sc/shuttle.jpg");
list[10].setAll(50290,"Puma football",5,800,"/sc/puma.jpg");
list[11].setAll(50475,"Mikasa volleyball",5,1200,"/sc/volley.jpg");
list[12].setAll(50778,"Wilson basketball",5,1500,"/sc/wilson.jpg");
}

```

ความแตกต่างระหว่าง

Part I และ Part II

- สำหรับ Part I จะมีการใช้งานอาร์เรย์แบบปกติ array[] และมีการใช้งานคู่กับอัลกอริทึมค่อนข้างมาก ส่วน Part II จะมีการนำเข้า Java.util.ArrayList มาใช้งาน ซึ่งแพ็คเกจนี้จะอำนวยความสะดวกในการใช้งานมากขึ้น มีการใช้งานอัลกอริทึมที่น้อยลง สั้นลง รวดเร็วขึ้น ทำให้ผู้ใช้เขียนโค้ดได้ง่าย และโปรแกรมทำงานได้ดียิ่งขึ้น
- การสร้างอาร์เรย์แบบปกติ จะบังคับให้ผู้ใช้เก็บข้อมูลได้ตามขนาดของมัน หากเต็มแล้วต้องการจะเพิ่มข้อมูลอีก จำเป็นต้องลบหรือสร้างอาร์เรย์ขึ้นมาใหม่ที่มีขนาดใหญ่กว่าเดิม ซึ่งเป็นปัญหาสำหรับไตรห้ายคน แพ็คเกจ Java.util.ArrayList นี้เองจึงถูกสร้างมาเพื่อแก้ปัญหาดังกล่าว จะทำให้อาร์เรย์ของเรามีความยืดหยุ่นมากขึ้น กล่าวคือ สามารถเพิ่มข้อมูลได้เรื่อยๆ นั่นเอง

DATA STRUCTOR AND ALGORITHM

Delete **Insert**

Code	66666
Name	Ari
Price	6000
Type	6
Instead No.	19

Insert

search code :

No.	Code	Name	Price	Type	Address
3	tu/85	Skechner s...	2500.0	1	/sc/ske.jpg
4	20096	Umbro Glo...	800.0	2	/sc/ub.jpg
5	20504	Nike footb...	1500.0	2	/sc/shirtNi...
6	30319	Ping stick...	1300.0	3	/sc/golfstic...
7	30873	Anyday golf...	5000.0	3	/sc/golfbag...
8	40152	Yonex strike	2400.0	4	/sc/yon.png
9	40197	Kawasaki ...	1700.0	4	/sc/kawa.jpg
10	40601	Victor shutt...	200.0	4	/sc/shuttle...
11	50290	Puma foot...	800.0	5	/sc/puma.j...
12	50475	Mikasa voll...	1200.0	5	/sc/volley.jpg
13	50778	Wilson ba...	1500.0	5	/sc/wilson...
14	11111	Fila	1000.0	1	/sc/home....
15	22222	Asics	2000.0	2	/sc/home....
16	33333	New Balan...	3000.0	3	/sc/home....
17	44444	Baoji	4000.0	4	/sc/home....
18	55555	Crocs	5000.0	5	/sc/home....
19	66666	Ari	6000.0	6	/sc/home....

Host : Krittaya001

Storage : -
Amount : 19

Present to Asst.Prof.DR.Chouvalit Khancombe
by.. Krittaya Tantichaiyakul 6705500269

3. เนื่องจากอาเรย์ชันด arraylist สามารถยืดหยุ่นได้ และมีคำสั่ง size() เพื่อตรวจสอบขนาดของอาเรย์ลิสต์ได้ เราจึงไม่จำเป็นต้องใช้ตัวแปร n อีกต่อไป เพียงแต่คำสั่งอาจพิมพ์ยาวนิดนึง เนื่องจากเราประ公示 arraylist ไว้ในไฟล์คลาส DataStore จึงต้องมีการเข้าถึง ก็คือ DataStore.list.size() แต่แลกกับการที่เวลาเรา insert delete หรืออัพเดทกระดานสต็อก เราไม่จำเป็นต้องกังวลเรื่องความถูกต้องของ n++ หรือ n— อีกแล้ว

```
int insertCode=Integer.parseInt(jTextField1.getText());
int insertIndex=Integer.parseInt(jTextField7.getText());
insertIndex-=1;
for(int i=0;i<DataStore.list.size();i++){
    if(insertCode==DataStore.list.get(i).code){
        System.out.println("This code exist already!");
        JOptionPane.showMessageDialog(this, "This code exist already");
        return;
    }
}
if(insertIndex<0){
```

4. ปกติเราจะสร้างตัวแปร UB ซึ่งเป็นขนาดของอาเรย์ แต่การนำแพ็กเกจมาใช้ อาเรย์ลิสของเราจะไม่มีวันเต็ม ดังนั้นตัวแปร UB จึงไม่จำเป็นสำหรับการใช้อัลกอริทึมอีกต่อไป เวลากำหนดสเกล เราจะสามารถตั้งว่า 0-DataStore.list.size() หรือพูดง่ายๆ ก็คือ ตั้งแต่ 0 ถึง จำนวนสมาชิกทั้งหมด ได้เลย

5. ในหน้า Menu คำสั่งปุ่ม next back ปกติเราจะต้องสร้างเงื่อนไขแบบเลือกอย่างใดอย่างหนึ่ง คือ ค่า current อยู่ภายใต้จำนวนสินค้าที่มีอยู่ใหม เพื่อใช้ว่าข้อมูลภายในสินค้าที่มีอยู่ ต่อมากาค่า current อยู่ภายใต้ขนาดของอาเรย์ใหม เพื่อใช้ว่าข้อมูลซ่องว่างที่เหลือด้วย แต่สำหรับการนำแพ็กเกจมาใช้ เราจะไม่มีเพดาน ดังนั้น ไม่จำเป็นต้องใช้ว่าตำแหน่งข้อมูลที่ว่างแล้ว สามารถทำให้ current ไม่เกินจำนวนสินค้าหรือ .size() ได้เลย

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    if(current<(DataStore.list.size()-1)){
        current++;
        jLabel1.setIcon(new javax.swing.ImageIcon(
        getClass().getResource(DataStore.list.get(current).pic)
        ));

    }else{
        JOptionPane.showMessageDialog(this, "Overflow!");
        System.out.println("Overflow");
    }
}
```

6. สำหรับการ insert ถือเป็นหัวข้อที่เน้นความแตกต่างระหว่างอาร์เรย์แบบปกติกับอาร์เรย์ลิสต์มากที่สุด ปกติการที่เราจะแทรกข้อมูลลงอาร์เรย์แบบปกติ เราจะใช้อัลกอริ듬มากรีมตั้งแต่ตรวจสอบว่าได้ดั้งๆ ใหม่ อาร์เรย์เต็มหรือยังและตำแหน่งที่ต้องการแทรกอยู่哪里ในอาร์เรย์ใหม่ ซึ่งสำหรับ arraylist เราไม่มีเพดาน จึงไม่จำเป็นต้องใช้ if ตรวจสอบว่าอาร์เรย์เต็มหรือยัง แทรกข้อมูลเข้าไปได้เลย ต่อมา if ตรวจสอบว่าตำแหน่งที่ต้องการแทรกเกินอาร์เรย์ลิสต์ใหม่ จากเดิม `insertIndex>=(UB-1)` เราสามารถแทรกลงไปตรงไปตำแหน่งไหนของอาร์เรย์ก็ได้ รวมถึงช่องว่างด้วย แต่ครั้งนี้เราจะไม่มีช่องว่างแล้ว จึงต้องเปลี่ยน UB เป็น `size()` และที่สำคัญ ผู้ใช้งานมีโอกาสที่จะแทรกข้อมูลเข้าไปเป็นตัวสุดท้ายได้ด้วย เราจะต้องขยายสเกล +1 โดยเปลี่ยนจาก `>= เป็น >` หรือให้ `size()+1` ก็ได้เหมือนกัน

```

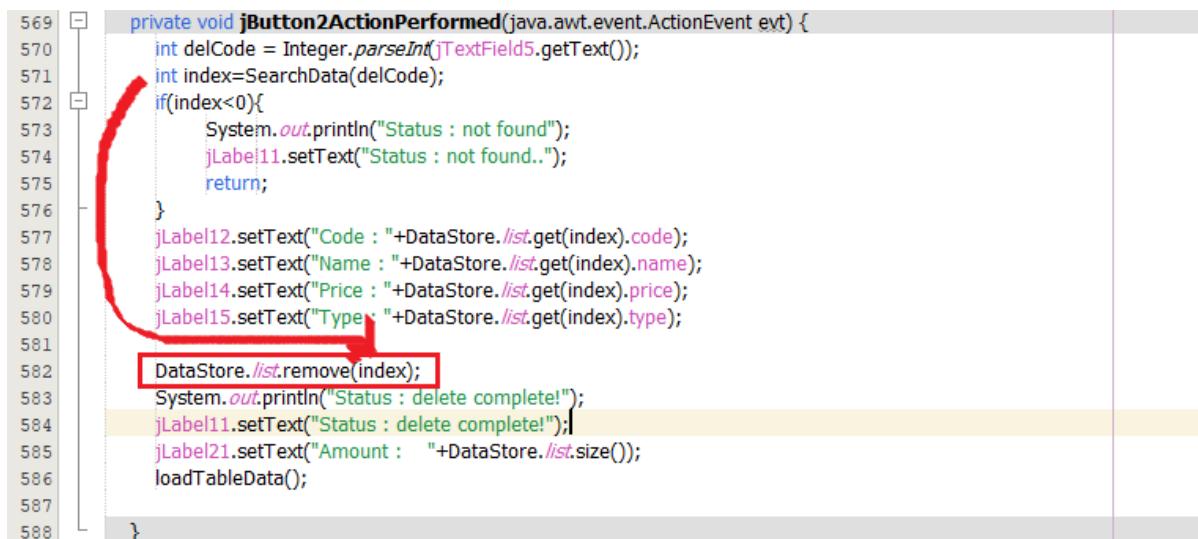
530 int insertCode=Integer.parseInt(jTextField1.getText());
531 int insertIndex=Integer.parseInt(jTextField7.getText());
532 insertIndex-=1;
533 for(int i=0;i<DataStore.list.size();i++){
534     if(insertCode==DataStore.list.get(i).code){
535         System.out.println("This code exist already!");
536         JOptionPane.showMessageDialog(this, "This code exist already!");
537         return;
538     }
539 }
540 if(insertIndex<0){
541     System.out.println("Underflow!");
542     JOptionPane.showMessageDialog(this, "Underflow!");
543     return;
544 }
545 if(insertIndex>(DataStore.list.size())){
546     System.out.println("Overflow!");
547     JOptionPane.showMessageDialog(this, "Overflow!");
548     return;
549 }
550 DataStore.Products obj2 = new DataStore.Products();
551 obj2.code=Integer.parseInt(jTextField1.getText());
552 obj2.name=jTextField2.getText();
553 obj2.price=Integer.parseInt(jTextField3.getText());
554 obj2.type=Integer.parseInt(jTextField4.getText());
555 obj2.pic="/sc/home.png";
556 DataStore.list.add(insertIndex,obj2);
557 System.out.println("Status : insert complete!");
558 jLabel21.setText("Amount : "+DataStore.list.size());
559 loadTableData();
560 }
561
}

```

Temp

การใช้ฟังก์ชัน add เป็นคำสั่งที่จะเพิ่มข้อมูลลงอาร์เรย์ลิสต์ตอนมาร์ต โดยที่เราไม่จำเป็นต้องขับอีลมีนตัวข้างหลังออกทุกตัวอีกแล้ว ซึ่งตรงนี้ถือว่าเป็นการลดขั้นตอนที่ดีมาก ตอบโจทย์วิชานี้ก็คือ การออกแบบอัลกอริทึมอย่างมีประสิทธิภาพ ทำให้จำนวน Big O ลดลง แต่คำสั่งนี้มีเงื่อนไขว่า ต้องแนบอาเกิร์มีนตัวที่ 2 เป็นอ้อมเจ็กที่เป็นข้อมูลใหม่ลงไปด้วย เราจะต้องสร้างอ้อมเจ็กใหม่สำหรับเซ็ตค่าที่ได้รับจาก TextField ไว้ก่อน หลังจากที่อ้อมเจ็กตัวนั้นมีหน้าตาของมันเรียบร้อยแล้ว แล้วจึงสามารถใส่ลงไปบนฟังก์ชัน add(int, object) เพื่อเพิ่มข้อมูลได้แล้ว

2. สำหรับการ delete จะมีคำสั่งที่อำนวยความสะดวกก็คือ remove() มาพร้อมใช้งาน เราไม่จำเป็นต้องใช้อัลกอ for แล้วขับตัวหลังมาแทนตัวหน้าอีกเช่นกัน ใช้คำสั่ง remove() ก็จะลบแต่คำสั่งนี้มีเงื่อนไขคือ ต้องใส่อากิร์มีนเป็นค่า index เพื่อลบตำแหน่งนั้นด้วย ดังนั้น เราจึงต้อง search หาตำแหน่งดังกล่าวก่อน แล้วจึงนำค่านั้นใส่ลงบนอาเกิร์มีนได้เลย เช่น remove(index)



```

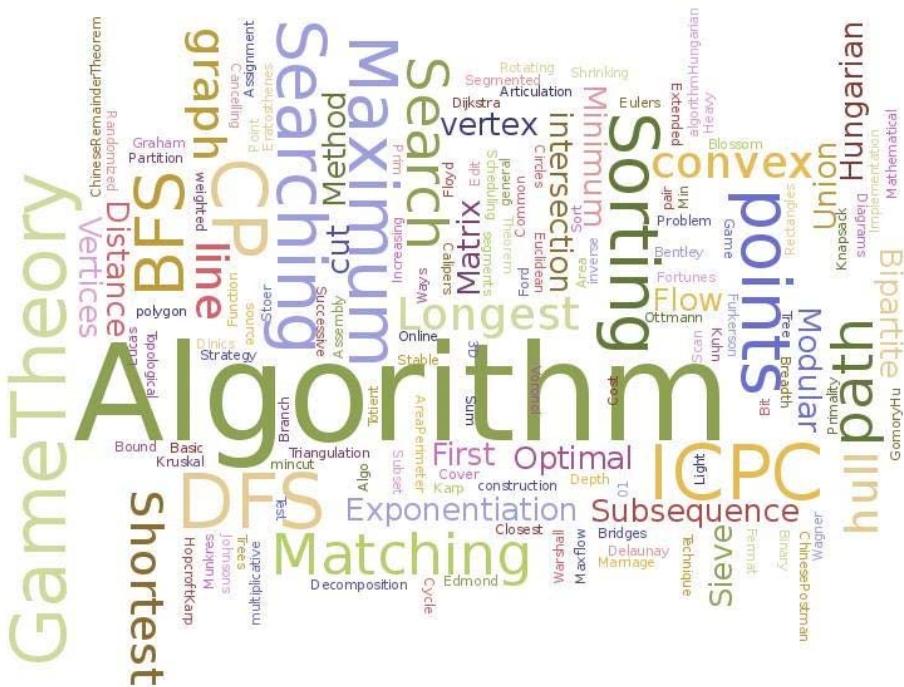
569 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
570     int delCode = Integer.parseInt(jTextField5.getText());
571     int index=SearchData(delCode);
572     if(index<0){
573         System.out.println("Status : not found");
574         jLabel11.setText("Status : not found..");
575         return;
576     }
577     jLabel12.setText("Code : "+DataStore.list.get(index).code);
578     jLabel13.setText("Name : "+DataStore.list.get(index).name);
579     jLabel14.setText("Price : "+DataStore.list.get(index).price);
580     jLabel15.setText("Type : "+DataStore.list.get(index).type);
581
582     DataStore.list.remove(index);
583     System.out.println("Status : delete complete!");
584     jLabel11.setText("Status : delete complete!");
585     jLabel21.setText("Amount : " +DataStore.list.size());
586     loadTableData();
587
588 }

```

3. สำหรับการค้นหา search นั้นยังไม่มีคำสั่งที่ค้นหาตำแหน่งโดยตรง จึงมีที่ใช้แทนกันได้บ้างก็คือ contains() แต่คำสั่งนี้จะให้ค่า Boolean ว่าเจอกหรือไม่เจอกค่านั้น แต่โปรแกรมของเราจำเป็นต้องใช้คำสั่งที่ค้นหาตำแหน่งด้วย เพื่อนำมาประยุกต์ใช้กับการลบและแทรกข้อมูล หนูจึงยังคงต้องใช้อัลกอเดิมในการค้นหาเดต้าอยู่ค่ะ

รวมถึงการ sort ด้วยเช่นกัน แพ็คเกจอาร์เรย์ลิสต์ยังไม่มีคำสั่งในการจัดเรียงข้อมูลค่ะ

ดังนั้นแล้ว จุดประสงค์ของอาจารย์ สำหรับพาร์ทนี้ก็คือ ให้นักศึกษาเรียนรู้โครงสร้างข้อมูลและอัลกอริทึมหลักๆรูปแบบ เพื่อทำให้เข้าใจโครงสร้างข้อมูลของอาเรย์ที่แท้จริง และเอกลักษณ์ของอัลกอแต่ละแพ็คเกจว่าทำงานอย่างไร โดยเด่นในด้านไหน แบบไหนทำให้ทำงานได้รวดเร็วหรือช่วยลดช่องโหว่ที่อาจทำให้เกิดความผิดพลาดได้มากกว่า ส่วนผลให้นักศึกษาสามารถเขียนโปรแกรมได้หลากหลายรูปแบบมากขึ้น บางอันใช้แทนกันได้ บางอันใช้แทนกันไม่ได้ ทำให้เราสามารถวิเคราะห์ได้ว่า โจทย์ปัญหาต่างๆ ควรเลือกอัลกอแบบใดที่เหมาะสมที่สุดมาใช้งาน



ขอบคุณค่ะ

