บทนำ

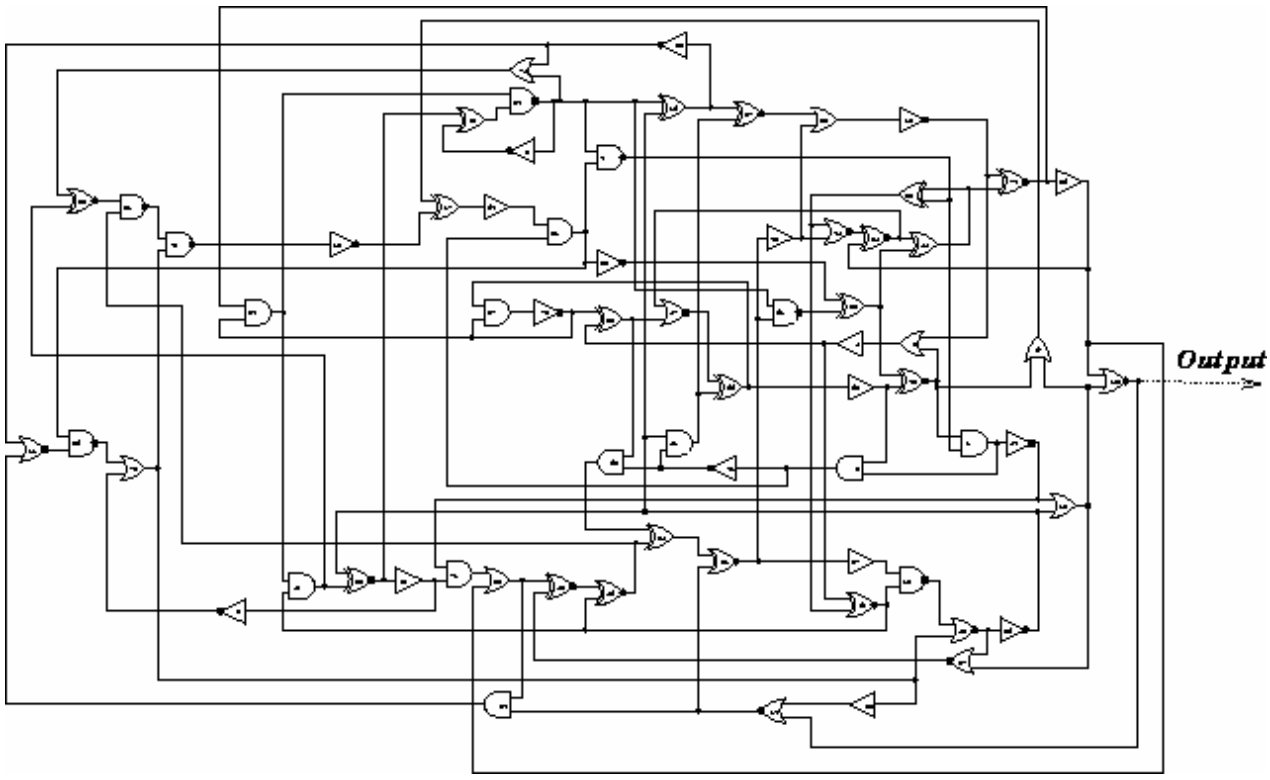
หัวข้อ

- นิยาม
- ตัวอย่างการใช้งาน
- วัตถุประสงค์ของการศึกษาโครงสร้างข้อมูล
- ตัวอย่างประสิทธิภาพเชิงเวลาการทำงาน

โครงสร้างข้อมูลคืออะไร

- วิธีการจัดเก็บและจัดการข้อมูลให้
  - ตรงตามความต้องการ
  - ทำงานรวดเร็ว
  - ประหยัดเนื้อที่
  - เข้าใจง่าย

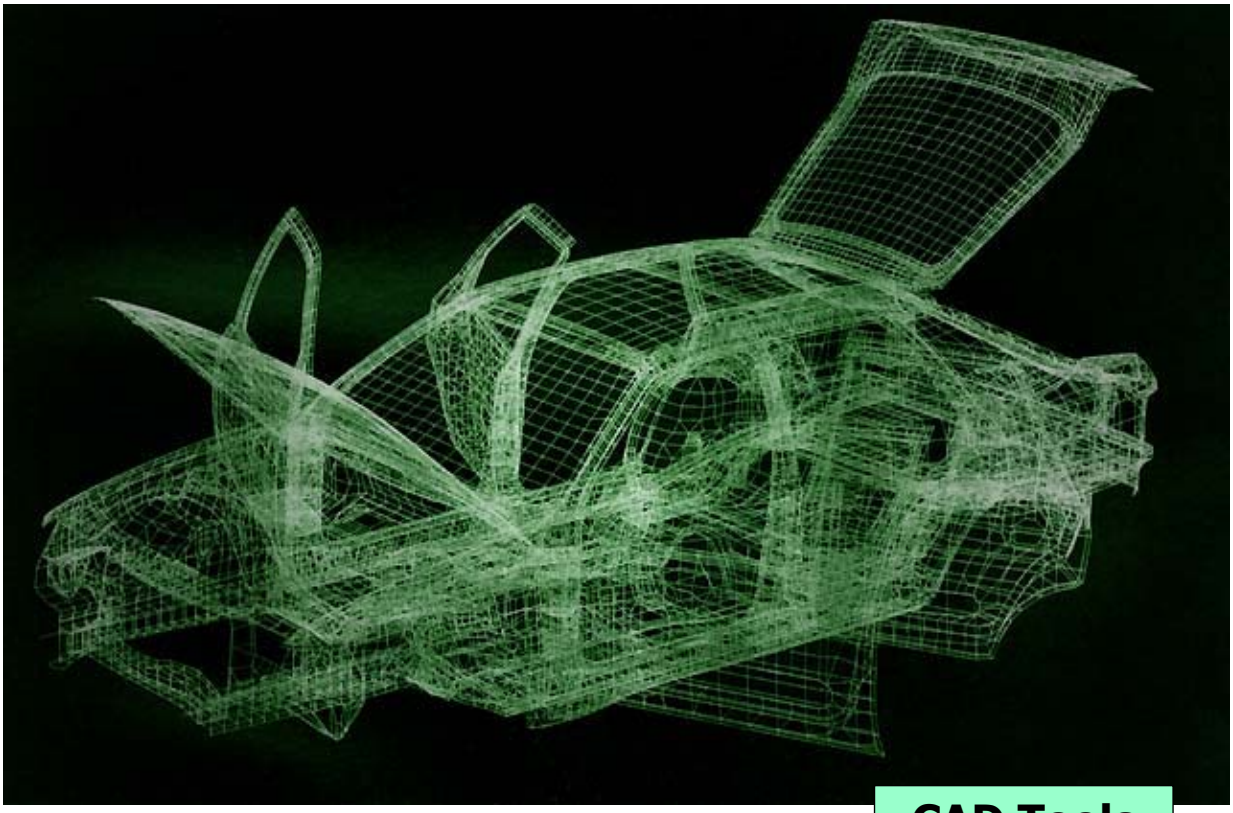
จะเขียนโปรแกรมจัดเก็บและจัดการอย่างไร ?



Logic Design Tools

VLSI Design

Games



CAD Tools

และอื่น ๆ

- Google เก็บเอกสารอย่างไร ทำให้ค้นได้รวดเร็ว
- jvm เก็บออปเจกต์ต่าง ๆ คลาสต่าง ๆ ตัวแปรต่าง ๆ thread ต่าง ๆ ไว้อย่างไร
- Word processor เก็บตัวอักษร คำ ข้อความ ย่อหน้า สูตร รูป และอื่น ๆ ในเอกสารภายในหน่วยความจำอย่างไร ขณะที่เรากำลังใช้งาน
- ...

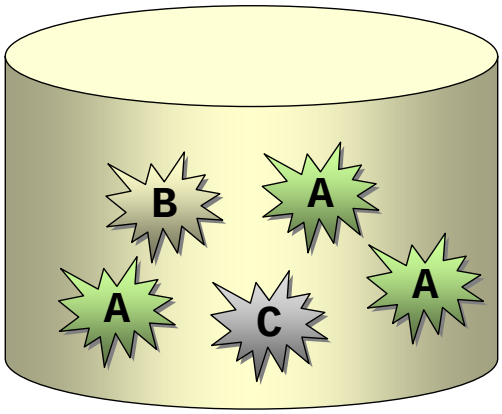
ล้วนเป็นปัญหาที่เกี่ยวกับโครงสร้างข้อมูล

ข้อมูลต่าง ๆ

- บางชนิด ตัวภาษามีให้แล้ว (primitive)
  - int, double, char, boolean, ...
- บางชนิดคลังคำสั่งของระบบมีให้ใช้ (class)
  - String, Color, BigDecimal, ArrayList, HashMap, ...
- และมีอีกมากมายที่ต้องออกแบบและสร้างเอง

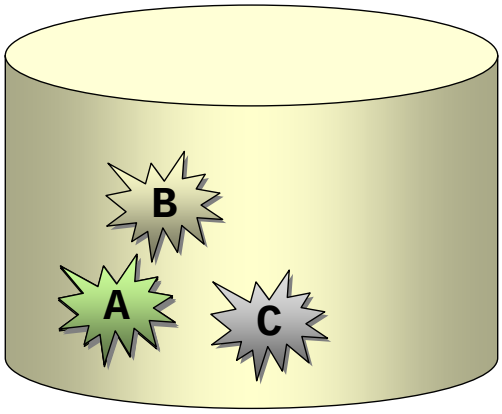
ที่เก็บข้อมูลแบบพื้นฐาน : Collection

- เก็บข้อมูลไม่มีอันดับ ซ้ำได้



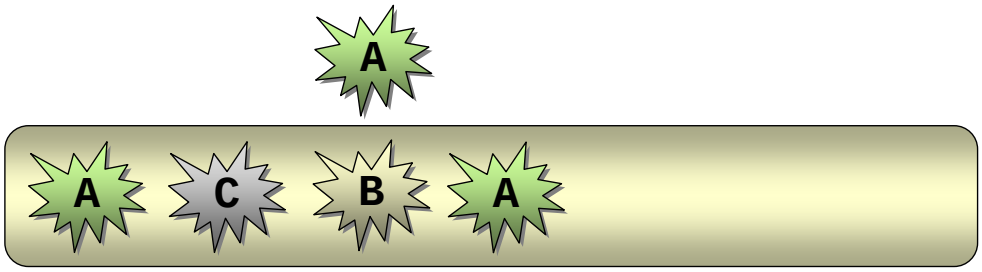
ที่เก็บข้อมูลแบบพื้นฐาน : Set

- เก็บข้อมูลไม่มีอันดับ ไม่ให้ซ้ำ



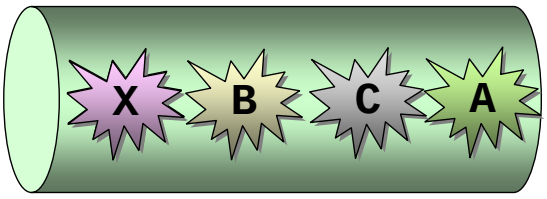
ที่เก็บข้อมูลแบบพื้นฐาน : List

- เก็บข้อมูลเรียงแบบมีอันดับ ซ้ำได้



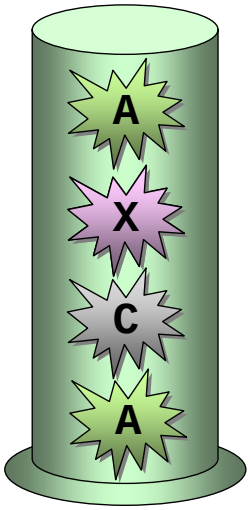
ที่เก็บข้อมูลแบบพื้นฐาน : Queue

- ข้อมูล เข้าก่อน ออกก่อน (First-In First-Out)



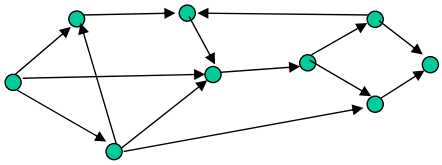
ที่เก็บข้อมูลแบบพื้นฐาน : Stack

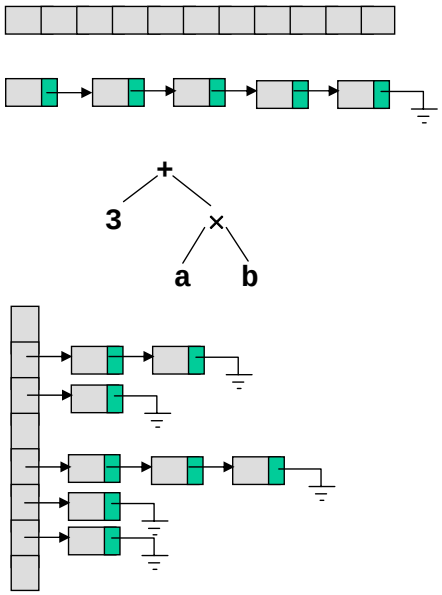
- ข้อมูล เข้าหลัง ออกก่อน (Last-In First-Out)



วิธีสร้างที่เก็บข้อมูล

- สร้างด้วยอาเรย์
- สร้างด้วยการโยง
- สร้างด้วยต้นไม้
- สร้างด้วยตาราง
- อื่น ๆ





วัตถุประสงค์

เลือกให้เป็น

ใช้ให้เป็น

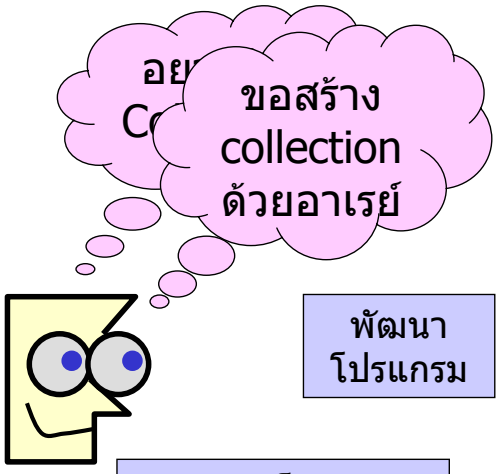
สร้างให้เป็น

เลือกให้เป็น

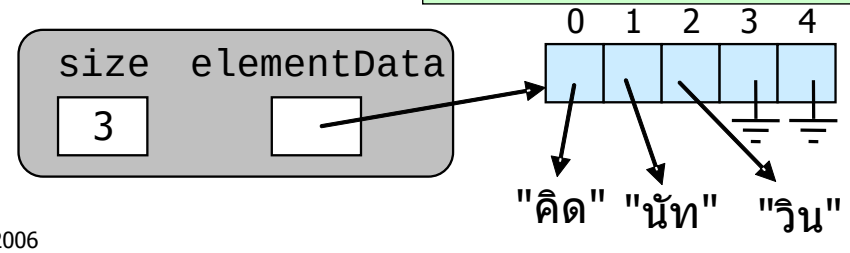
ใช้ให้เป็น

```
public static PuzzleBoard solve(PuzzleBoard b) {
  Set set = new ArraySet();
  Queue queue = new ArrayQueue();
  queue.enqueue(b); set.add(b);
  while ( !queue.isEmpty() ) {
    b = queue.dequeue();
    for (int d = 0; d < 4; d++) {
      PuzzleBoard b2 = b.moveBlank(d);
      if (b2 != null) {
        if ( b2.isAnswer() ) return b2;
        if (! set.contains(b2) ) {
          queue.enqueue(b2); set.add(b2);
        }
      }
    }
  }
  return null;
}
```

สร้างให้เป็น



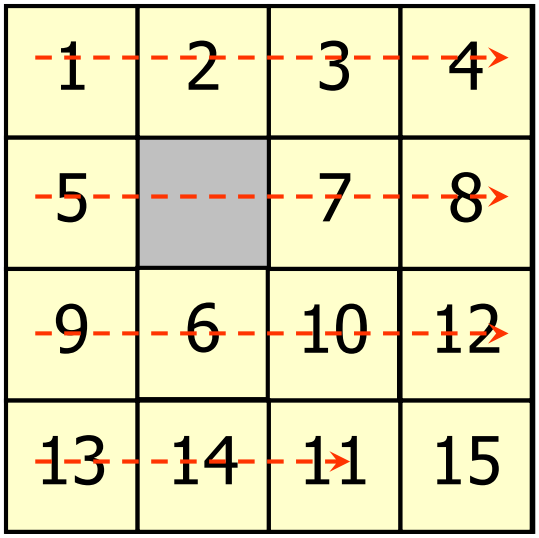

```
public class ArrayCollection
       implements Collection {
 private Object[] elementData;
 private int      size;

 public ArrayCollection(int c) {
   elementData = new Object[c];
   size = 0;
 }
 public void add(Object e) {
   elementData[size++] = e;
 }
 public boolean size() {
   return size;
 }
 public int isEmpty() {
   return size == 0;
 }
 ...
```



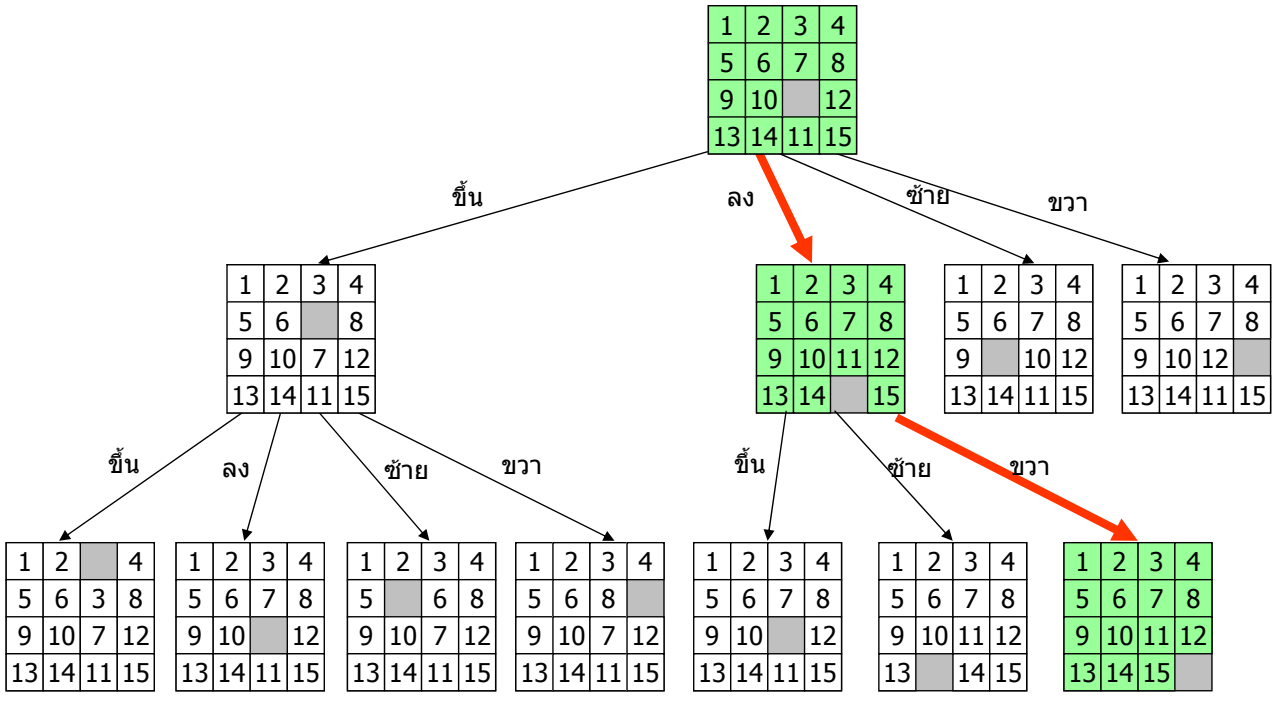
โครงสร้างข้อมูลที่ดี

- ให้บริการตามที่ต้องการ
- ใช้ง่าย
- กินเนื้อที่น้อย
- ทำงานได้อย่างรวดเร็ว

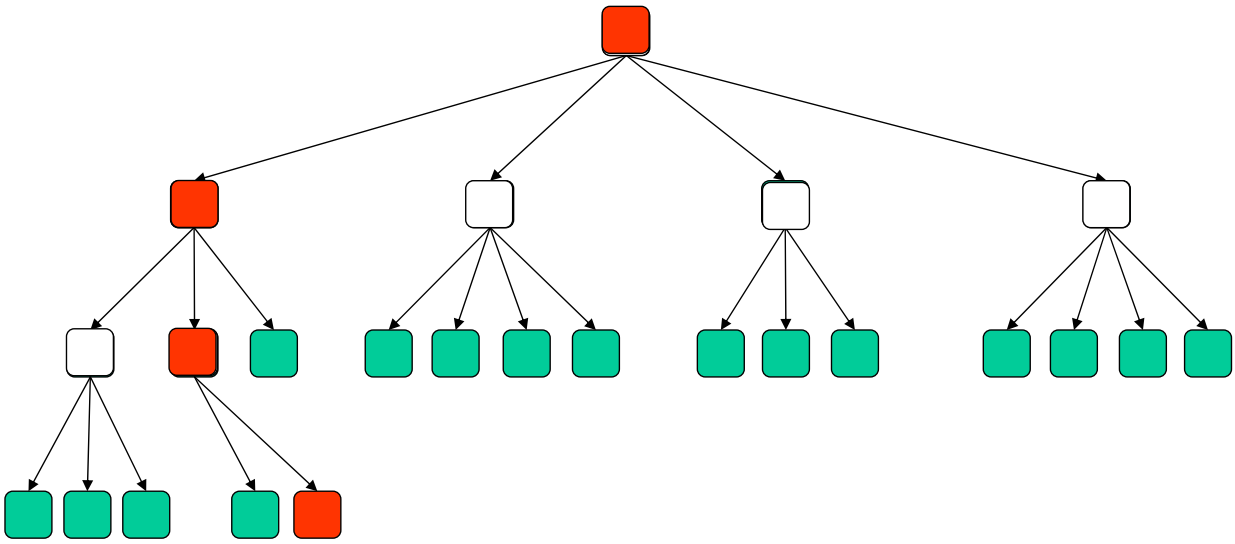
ตัวอย่าง : 15-puzzle



พบคำตอบ พบวิธีเลื่อน



ใช้ Queue เก็บตาราง

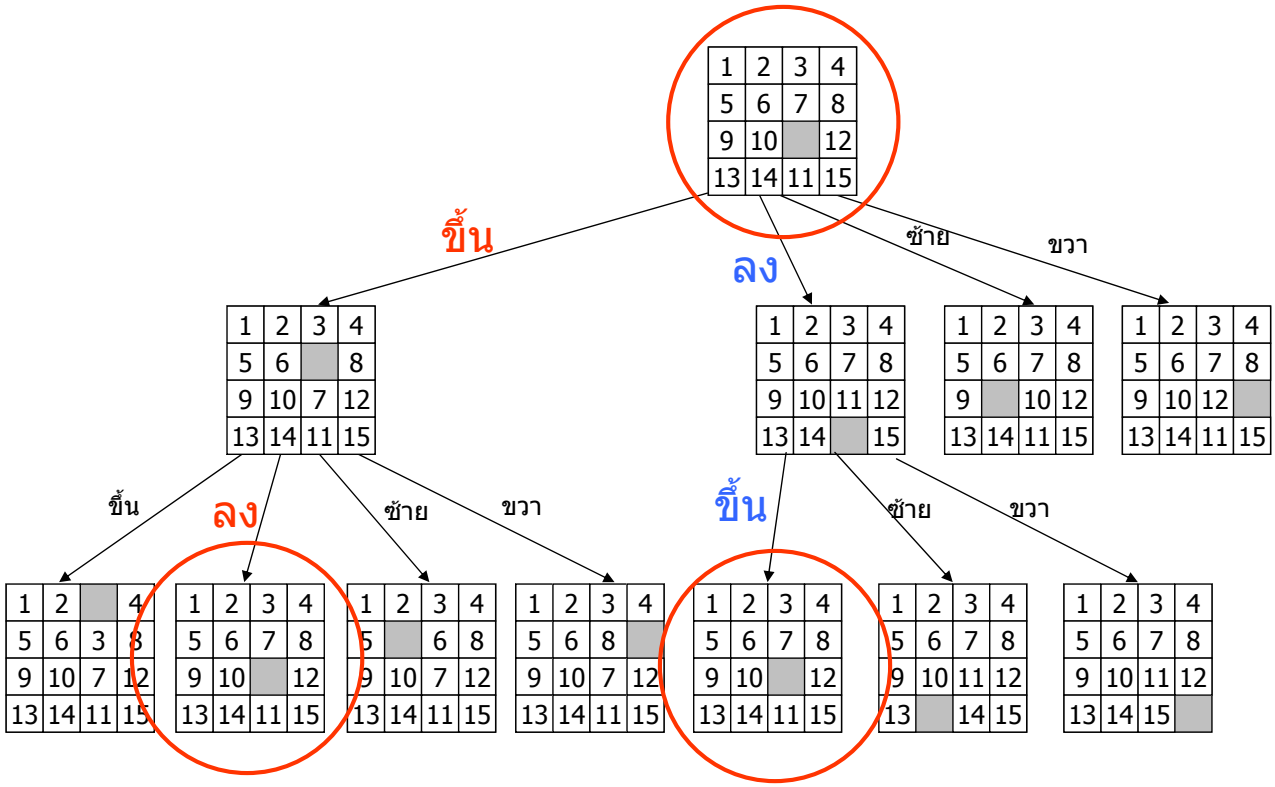


15-puzzle : ส่วนของโปรแกรม

```
public static PuzzleBoard solve(PuzzleBoard b) {

  Queue queue = new ArrayQueue();
  queue.enqueue(b);
  while ( !queue.isEmpty() ) {
    b = queue.dequeue();
    for (int d = 0; d < 4; d++) {
      PuzzleBoard b2 = b.moveBlank(d);
      if (b2 != null) {
        if ( b2.isAnswer() ) return b2;

        queue.enqueue(b2);

      }
    }
  }
  return null;
}
```

ใช้ queue เก็บตารางที่ผลิตใหม่

15-puzzle : ตารางที่ผลิตอาจซ้ำกัน



15-puzzle : ส่วนของโปรแกรม

```
public static PuzzleBoard solve(PuzzleBoard b) {
  Set set = new ArraySet();
  Queue queue = new ArrayQueue();
  queue.enqueue(b);  set.add(b);
  while ( !queue.isEmpty() ) {
    b = queue.dequeue();
    for (int d = 0; d < 4; d++) {
      PuzzleBoard b2 = b.moveBlank(d);
      if (b2 != null) {
        if ( b2.isAnswer() ) return b2;
        if ( ! set.contains(b2) )  {
          queue.enqueue(b2);  set.add(b);
        }
      }
    }
  }
  return null;
}
```

ใช้ queue เก็บตารางที่ผลิตใหม่

ใช้ set เพื่อตรวจสอบความซ้ำซ้อน

15-puzzle : ผลการทดลอง

ตารางเริ่มต้น	จำนวนตารางที่ผลิต	เวลาการทำงาน (วินาที)			
		ArraySet	BSTSet	AVLSet	HashSet
แบบที่ 1	552	0.03	0.02	0.04	0.05
แบบที่ 2	5242	1.94	0.22	0.18	0.12
แบบที่ 3	132049	1819.6	7.08	5.71	2.56

สรุป

- ศึกษาวิธีการจัดเก็บและการจัดการข้อมูล
  - ตรงตามความต้องการ
  - ทำงานรวดเร็ว
  - ประหยัดเนื้อที่
  - เข้าใจง่าย
- เราต้อง
  - เลือกให้เป็น
  - ใช้ให้เป็น
  - สร้างให้เป็น