

Array CRUD

***Array* Introduction**

What is Array???

- เป็น Data Structure ที่ basic ที่สุด
- Array คือข้อมูลชนิดเดียวหรืออย่างน้อยขนาดเท่ากันที่เรียงต่อกันบน memory
- ใน Java จะเก็บข้อมูลไว้ใน Heap(Dynamic memory) เพราะเป็น object

Array Operation

- Random access (retrieve/update)
- Add/Insert unordered/ordered array
- Search in unordered/ordered array
- Delete unordered/ordered array

Array Add & Error Checking

Adding Data to An Array

- เราจะใส่ข้อมูลใหม่ลงใน Array โดยไม่สนใจ order/ลำดับ
- ที่ที่ดีที่สุดในการใส่ข้อมูลใหม่ลงใน Array คือ หลังสุด เพราะจะได้ไม่ไปแตะข้อมูลเก่าๆหรือย้ายข้อมูลเก่า

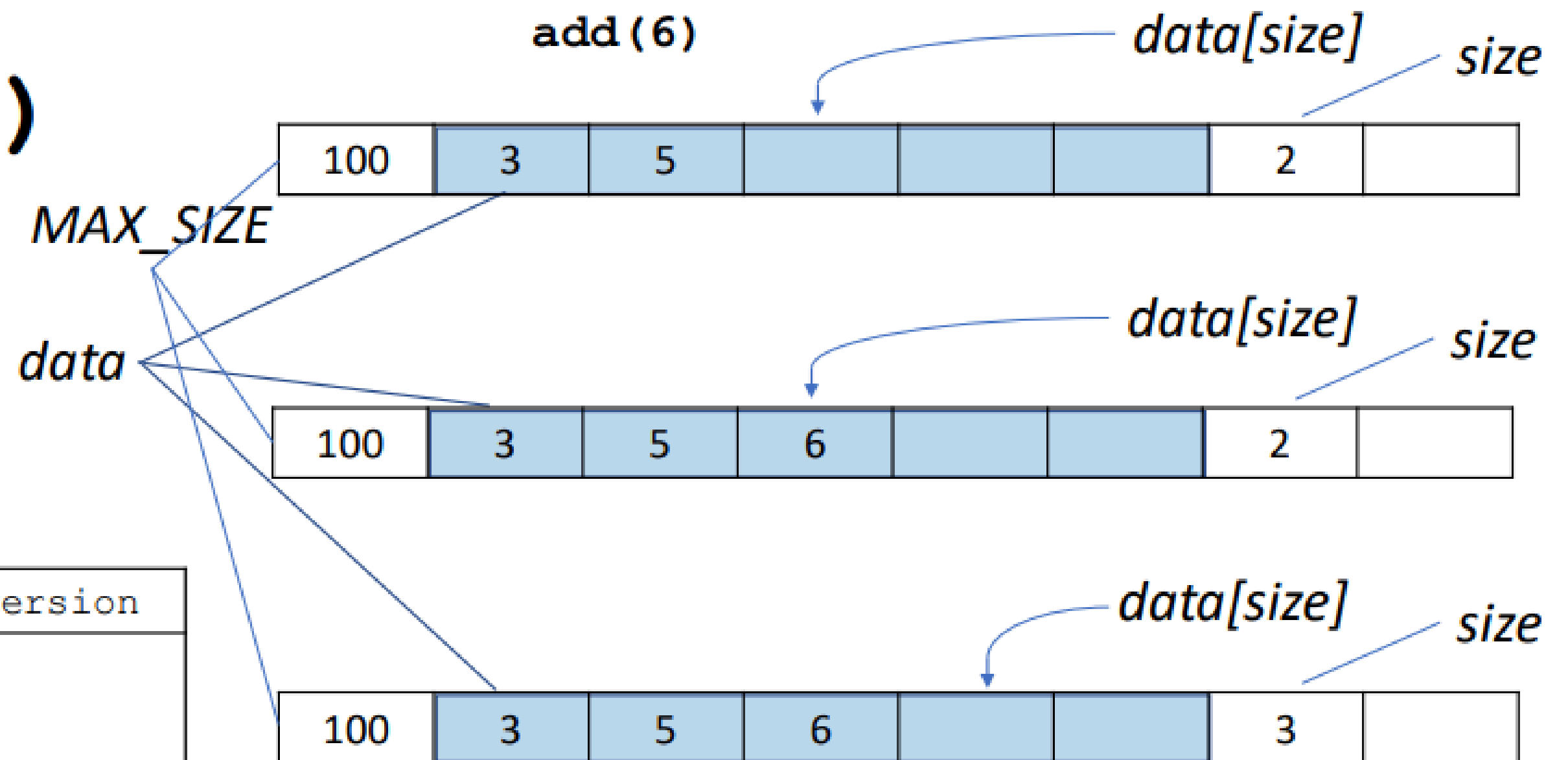
Method `add()`

Method `add()` version 1

```
public void add(int d) {  
    data[size]=d;  
    size=size+1;  
}
```

Method `add()` more compact version

```
public void add(int d) {  
    data[size++]=d;  
}
```



Note that there is no checking if the array is full.

Same method add(), change the main

- If you really need performance, you can leave add() as is and do the checking yourself.
- fast(?) but need to worry programmers

ArrayTester.java

```
public class ArrayTester {  
    public static void main(String args[]) {  
        MyArray mArray = new MyArray();  
  
        if (!mArray.isFull())  
            mArray.add(5);  
  
        System.out.print(mArray.toString());  
    }  
}
```


Method add() with checking for full

Method add() revised

```
public int add(int d) {  
    if(isFull()) return -1;  
    data[size++]=d;  
    return size;  
}
```

- ถ้า array ของคุณเต็มเมื่อไหร่ จะ return ค่า -1
- ถ้า array ยังไม่เต็มจะ return ค่า size
- ในบางครั้งการ return ตำแหน่งที่เก็บไว้อาจมีประโยชน์มากกว่า
- หรืออาจ return ค่า boolean เช่นถ้า add() สำเร็จ return true แต่ถ้าไม่ return false

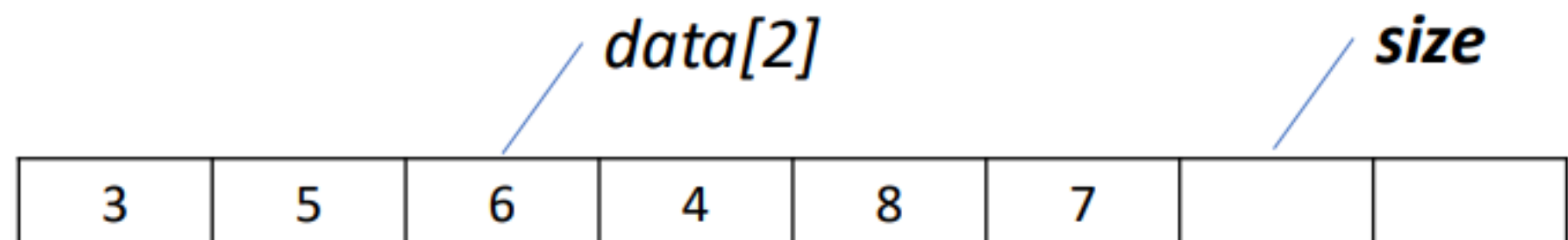
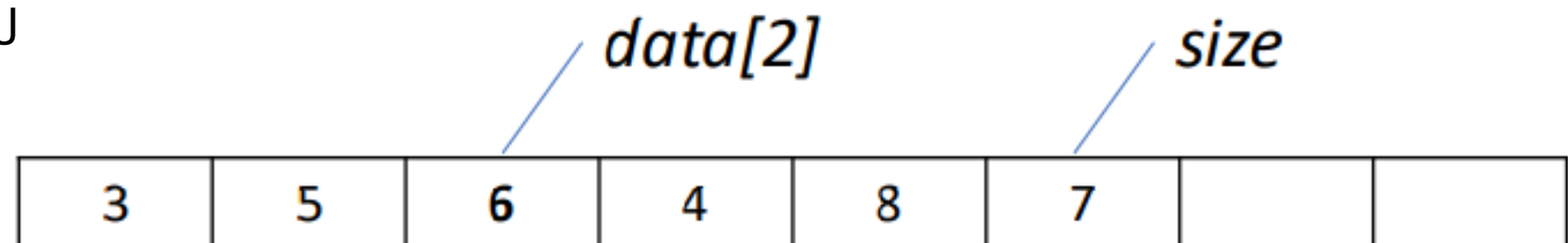
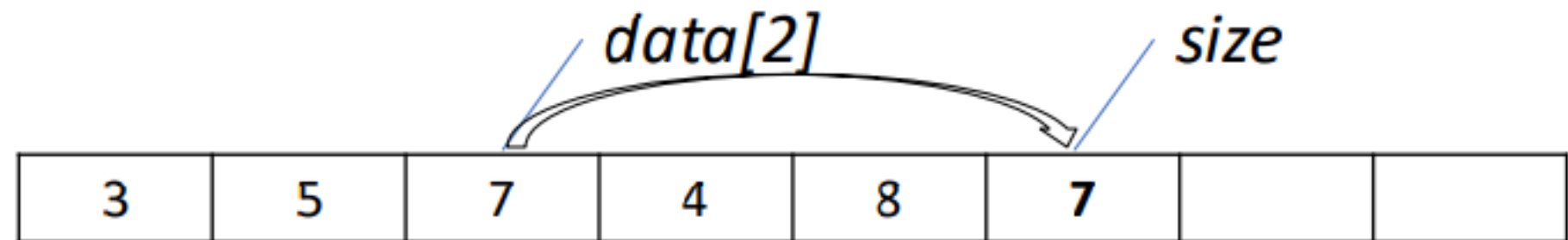
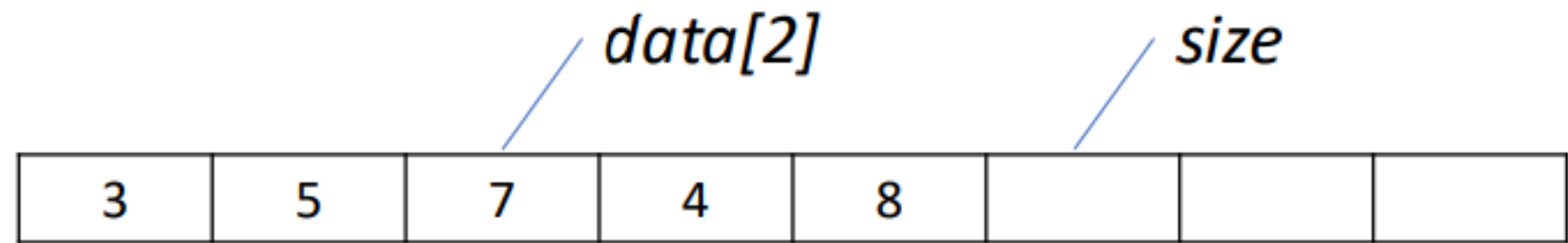
Note: การใช้ size==MAX_SIZE แทนการใช้ isFull() ทุกที่ในโค้ดไม่ใช่เรื่องที่ดี

Arrays Insert & Find

Array Insertion

- สมมติว่าเราต้องการแทรก 6 ใน index ที่ 2
- ถ้าเกิดไม่สนลำดับของข้อมูล
 - 1.ย้าย data[2] ไปไว้ข้างหลังสุดของ array
 - 2.นำ 6 ไปใส่ใน data[2] แทน 7 ที่ถูกย้ายไป
 - 3.ขยับ size ของ array ออกไป

Note: ขั้นตอนที่ 1 ต้องทำก่อนขั้นตอนที่ 2 เสมอ
แต่ ขั้นตอนที่ 3 ทำตอนไหนก็ได้



```
public void insert(int d, int index) {  
    data[size++] = data[index];  
    data[index] = d;  
}
```

Big-O = O(1)

Method insert(int d, int index)

```
public void insert(int d, int index) {  
    for(int i=size; i>index; i--) {  
        data[i] = data[i-1];  
    }  
    data[index] = d;  
    size++;  
}
```

Work, but bad practice!

```
public void insert(int d, int index) {  
    for(int i=size++; i>index; i--) {  
        data[i] = data[i-1];  
    }  
    data[index] = d;  
}
```

วิธีที่ 2
ทำงานยาก
และ debug
ยาก

Let find()

-สำหรับ array ที่ไม่สนใจลำดับ เราสามารถใช้วิธีที่ดีที่สุดคือ linear search(ค้นทีละตัว)

Simple implementation

```
public int find(int d) {  
    int index=-1;  
    for(int i=0; i<size; i++) {  
        if(data[i]==d) {  
            index = i;  
            break;  
        }  
    }  
    return index;  
}
```

Compact version

```
public int find(int d) {  
    for(int i=0; i<size; i++) {  
        if(data[i]==d) {  
            return i;  
        }  
    }  
    return -1;  
}
```

-Big-0 จะคล้ายกับ การแทรก array แบบสนใจลำดับ

Arrays Delete & Expand

Method delete(int index)

```
public void delete(int index) {  
    for(int i=index; i<size-1; i++) {  
        data[i] = data[i+1];  
    }  
    size--;  
}
```

ไม่แนะนำให้ใช้แบบที่ 2 (ไม่ควรทำ)

Very tempting, but again, bad practice!

```
public void delete(int index) {  
    for(int i=index; i<--size; i++) {  
        data[i] = data[i+1];  
    }  
}
```

THANK YOU

