

# 基礎資料結構與演算法

TYIC 桃高資訊社

# 資料結構與演算法

資料結構與演算法

(Data Structure & Algorithm, 簡稱 DSA)

在程式設計中有著非常重要的地位

使用好的資料結構和演算法

可能會使程式的執行速度變得更快

而使用不妥當的資料結構和演算法

則可能會使程式的執行速度變得緩慢

# 尋找最大、最小值

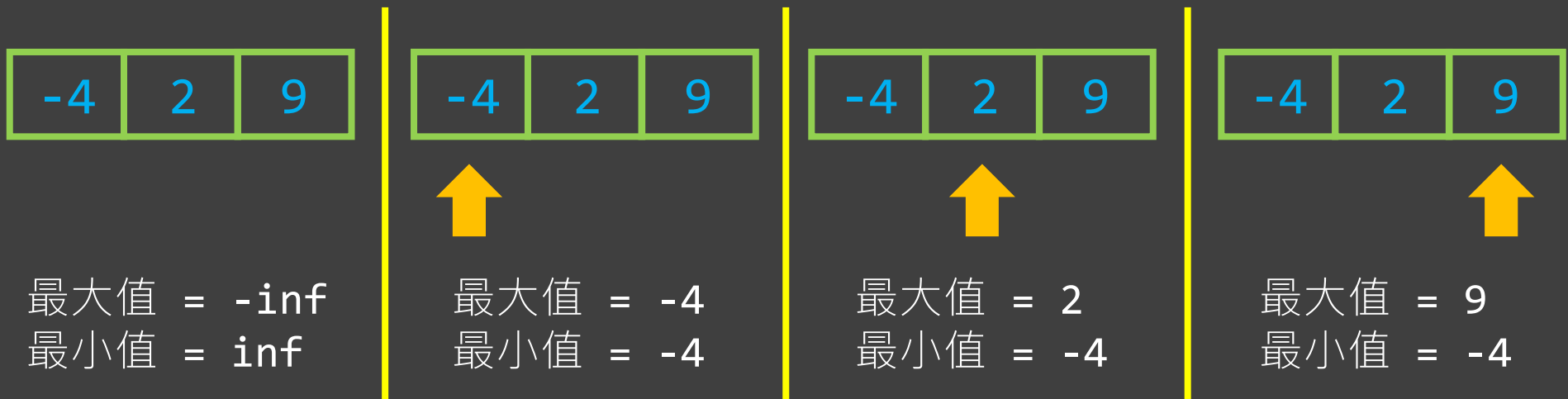
對於多個值，想要找尋最大、最小值

除了對資料排序外，也可利用以下方法：

依序讀取每個值，若較當前的最大值大或最小值小  
則將最大值或最小值變為該值

特別注意，最大值須初始化成比所有可能值小的數

最小值須初始化成比所有可能值大的數



# 尋找最大、最小值

```
import java.util.Scanner;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        int n = scanner.nextInt();
```

```
        int max = -2147483648, min = 2147483647;
```

```
        for (int i = 0; i < n; i++) {
```

```
            int p = scanner.nextInt();
```

```
            if (p > max) max = p;
```

```
            if (p < min) min = p;
```

```
        }
```

```
        System.out.printf("max = %d, min = %d", max, min);
```

```
    }
```

```
}
```

```
10
```

```
-1 5 -9 8 1000 2 -1999 2 0 1
```

```
max = 1000, min = -1999
```

```
console
```



```
java
```

# 獲取一正整數位數

若一正整數  $n$  滿足  $10^n \leq x = a \times 10^n < 10^{n+1}$  ( $1 \leq a < 10$ )

則  $\log(10^n) = n \leq \log(x) = n + \log(a) < \log(10^{n+1}) = n + 1$

又  $0 \leq \log(a) < 1$ ，得  $[\log(x)] = n$  (註： $[m]$  為下取整函數，如  $[2.7] = 2$ )

又已知  $10^n$  為  $n + 1$  位數，故  $x$  為  $n + 1 = [\log(x)] + 1$  位數

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        System.out.printf("%d has %d digit(s).", n, (int) Math.Log10(n) + 1);
    }
}
```

900999

900999 has 6 digit(s). console

123

123 has 3 digit(s). console



java

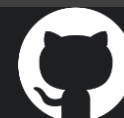
# 獲取一正整數之每一位數

末位數字即為該正整數除以 **10** 的餘數

該正整數除以 **10** 的商即為去除末位數字後的其他位數字

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        while (n != 0) {
            System.out.println(n % 10);
            n /= 10;
        }
    }
}
```



12345		114514	
5		4	
4		1	
3		5	
2		4	
1		1	
1	console	1	console

java