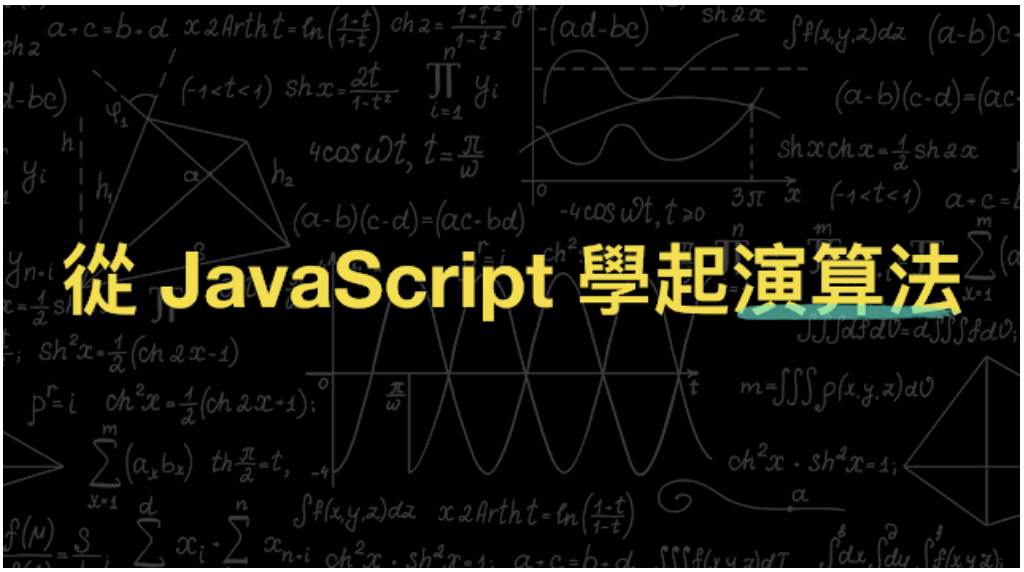
[演算法] Big O Notation & Time Complexity

9月 19, 2017 Algorithm , JavaScript , learning algorithm in javascript from scratch , Udemy



此系列筆記主要依照
[Udemy] Learning Algorithms in JavaScript from Scratch by Eric Traub 的課程脈絡加以整理，但部分程式碼是消化後以自己較易理解的方式重新撰寫，因此和原課程內容有些出入。

Big O Notation & Time Complexity

同樣的問題可以用許多種不同的方式加以解決，因此，我們需要一些指標來評量各種方式的好壞。在演算法中，常會使用 `Big O Notation` 和 `Time Complextiy` 來衡量一個演算法（函式）的好壞。通常，會根據這個函式隨著輸入的資料量增加時，執行時間會拉長多少來作為衡量的標準之一，下面會說明其中四種類型：

> 補充：
> Big O Notation 代表演算法時間函式的上限（Upper bound），表示在最壞的狀況下，演算法的執行時間不會超過Big-O。
> — [資料結構]演算法評估與資料型別

Constant Run Time (`O(1)`)

第一個類型是屬於 constant run time（ `O(1)` ），這個演算法（函式）的執行時間不會隨著輸入資料量的增加而增加。

以下面的函式為例，不論我們代入的資料量有多大，它都只是輸出陣列中第一和第二個元素的值，因此執行時間不會隨著輸入資料量的增加而增加。

```
let arr1 = [1,2,3,4,5]
let arr2 = [1,2,3,4,5,6,7,8,9,10]

/**
 * Constant Run Time：不會隨著輸入的資料量越大而使得執行時間變長
 * Big O Notation: "O(1)"
 **/
function log (arr) {
  console.log(arr[0])
  console.log(arr[1])
}
```

```
log(arr1)    // 1, 2
log(arr2)    // 1, 2
```

Linear Run Time (O(n))

下面的函式，當我們輸入的資料越多的時候，它就會需要等比例輸出越多的內容給我們，因此會需要消耗等比例越多的時間：

```
/**
 * Linear Run Time: 隨著資料量的增加，執行時間會等比增加
 * Big O Notation: "O(n)"
 **/
function logAll(arr) {
  for (let item of arr) {
    console.log(item)
  }
}

logAll(arr1)  // 1, 2, 3, 4, 5
logAll(arr2)  // 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

Exponential Run Time (O(n^2))

隨著資料量的增加，執行時間會以指數成長。以下面的函式為例，當我們輸入的陣列包含 5 個元素的時候，它會輸出 25 (5^2) 筆資料；但是當我們數入的陣列包含 10 個元素的時候，它則會輸出 100 (10^2) 筆資料：

```
/**
 * Exponential Run Time:  隨著資料量的增加，執行時間會誇張的增長
 * Big O Notation: "O(n^2)"
 **/
function addAndLog (arr) {
  for (let item of arr) {
    for (let item2 of arr) {
      console.log ('First', item + item2)
    }
  }
}
addAndLog(arr1)  // 25 pairs logged out
addAndLog(arr2)  // 100 pairs logged out
```

Logarithmic Run Time (`O(log n)`)

隨著資料量增加，執行時間雖然會增加，但增加率會趨緩。下面的程式碼類似 `findIndex` 的函式，當輸入的資料有 5 個元素時，它會先切對半後，再尋找，再切半再尋找，因此雖然隨著資料量增加，執行的時間會增加，但是當資料量越大時，執行速度增加的情況越不明顯：

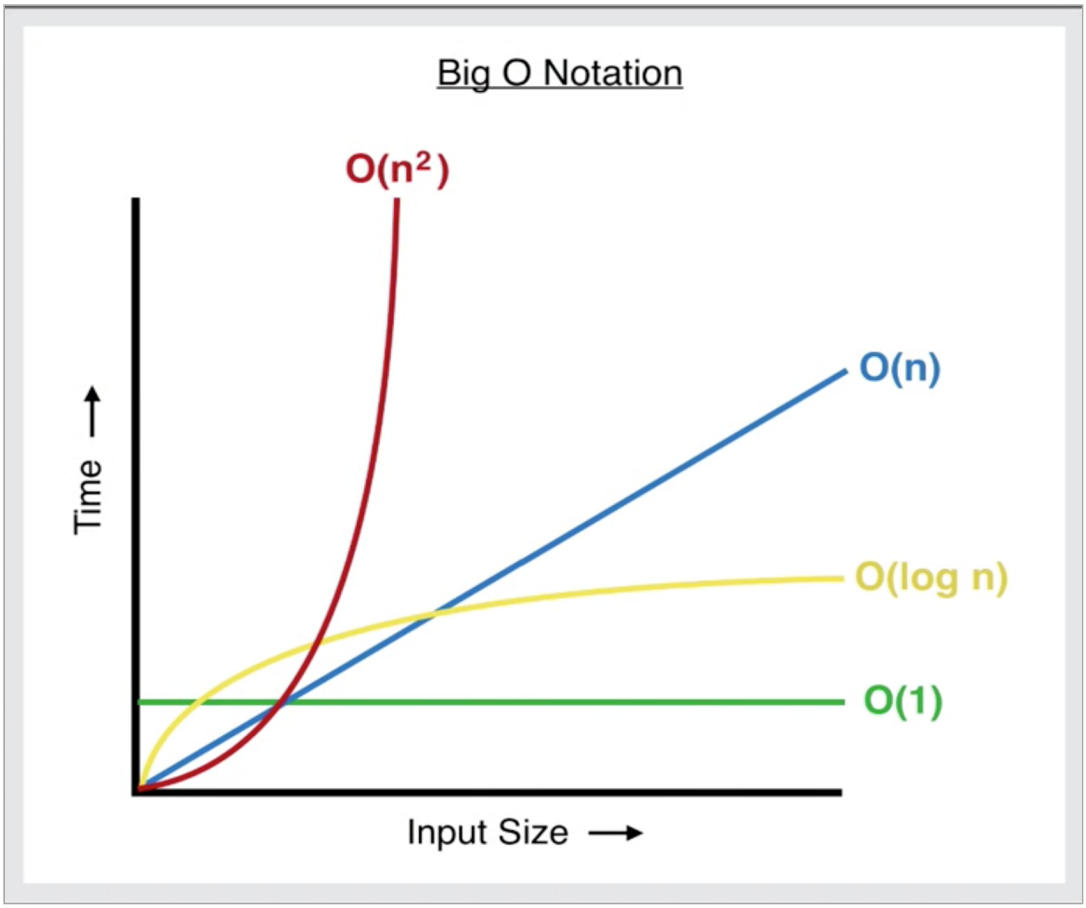
```
/**
 * Logarithmic Run Time: 隨著資料量增加，執行時間雖然會增加，但
 * Big O Notation: "O (log n)"
 **/
function binarySearch (arr, key) {
  let low = 0
  let high = arr.length - 1
  let mid
  let element

  while (low <= high) {
    mid = Math.floor((low + high) / 2, 10)
    element = arr[mid]
    if (element < key) {
      low = mid + 1
    } else if (element > key) {
      high = mid - 1
    } else {
      return mid
    }
  }
  return -1
}

console.log(binarySearch(arr1, 3))
console.log(binarySearch(arr2, 3))
```

圖示

把上面這四種類型用圖線表示，縱軸是時間、橫軸是輸入資料量的多少，可以用來判斷這四種類型的演算法（函式）的好壞：



> 圖片來源：Learning Algorithms in JavaScript from Scratch by Eric Traub @ Udemy

完整程式碼

```
      element = arr[mid]
      console.log('ele', mid, element)
      if (element < key) {
        low = mid + 1
      } else if (element > key) {
        high = mid - 1
      } else {
        return mid
      }
    }
    return -1
  }

  console.log(binarySearch(arr1, 1))
```

```
console.log(binarySearch(arr2, 1))
```

- Big O Notation @ Repl.it

資料來源

- Learning Algorithms in JavaScript from Scratch by Eric Traub @ Udemy

Share:

Related Posts:

[演算法] Is Palindrome：判斷順寫逆寫是不是一樣

[演算法] Harmless Ransom Note：計算陣列中各元素出現的次數

[演算法] Caesar Cipher: 往後或往前推移英文字母

[演算法] Reverse Array in Place：暫存變數的使用

[演算法] 計算平均、中位數、眾數（Mean Median Mode）

hannahpun • 2 years ago

謝謝你的分享~~

回复 • 分享 ›

pjchender hannahpun • 2 years ago

你的部落格很好看耶！謝謝分享！

回复 • 分享 ›

在 PJCHENDER 愛分享 上还有

[學習筆記目錄] 你所不知道的 JavaScript: Understanding the Weird

6条评论 • 2 years ago

Po-Jung Chen —

很高興對你有幫助

[演算法] Max Stock Profit ~ PJCHENder那些沒告訴你的小細節

2条评论 • 2 years ago

Po-Jung Chen —

哇！真的精簡非常多耶！感謝提供！！

```
var maxProfit = function(prices) {
  let curMax = 0;
  let maxSoFar = 0;

  for (var i = 1; i < prices.length; i++) {
    curMax = Math.max(0, (curMax += prices[i]
    maxSoFar = Math.max(curMax, maxSoFar
  }
  return maxSoFar;
};
```

[React] 搭配 React Router 打造一個動態麵包屑（dynamic breadcrumb)

關於「自學」這件事 ~ PJCHENder那些沒告訴你的小細節

Tags

Udemy PHP CSS 數位生活 HTML MySQL jQuery APP分享 心理健康 操控人心 行為改變 理財 研究生入門 WORD 開箱文

熱門文章

[多圖] 三星 Samsung S10+ Plus 與 Note 10 選擇困難？相機拍照、螢幕與效能比較心得

keywords: samsung , camera , s10+ , s10plus , note10 , performance , screen 最近剛好有機會同時入手三星（Samsung）的 S10+ Plus 和 Note 10（不是 Note 10+），...

[教學] PHP & MySQL 學習筆記 第八堂（資料庫建置：phpMyAdmin資料庫建置並呈現於網頁）

非常抱歉，關於部落格內 PHP 的部分目前已經停止維護，因本人已經很久沒有寫 PHP ，且文章中所使用的 PHP 版本偏舊，希望有心學習 PHP 的朋友們，可以參考 Codecademy 的課程，或近一步嘗試 Laravel 這個 PHP 框架（可透過 laracasts...

精選文章

[學習筆記目錄]列文章

► 四月 (1)

► 三月 (3)

► 二月 (1)

► 一月 (14)

► 2016 (67)

► 2015 (100)

► 2014 (32)