

演算法- 效能分析



十八豆教育科技

首頁 教學平台 使用手冊 個人購書 教學問題 教學策略 系統內容 學校採購 聯絡我們 舊版網頁

精準教育-

DICE程式設計精準教學平台

我是... 現在用戶 家長,學生 平台理念 教師帳號

解決一個問題，
有數個方法，
例如從嘉義到
成都

方法1

- 水上機場(飛機)→金門(船)→廈門(飛機)→成都

方法2

- 嘉義(汽車)→高雄(飛機)→香港(飛機)→成都

方法3

- 嘉義(汽車)→桃園(飛機)→成都

我們可以從時間、價格，
甚至有趣性去評估與比較這三種演算法

使用電腦執行演算法，如何評估效能？

空間複雜度

Space Complexity

執行完畢所需的記憶體空間

時間複雜度

Time Complexity

執行完畢所需的時間

時間複雜度

用一個故事來說明，

“有個大臣因功向國王邀賞，獎賞內容是要求國王將第一個棋盤格子中擺 1 粒稻米、第二個棋盤格子中擺 2 粒稻米、第三個格子中擺 4 粒... 以此類推到 101 個格子。

國王表示：安安，這個要求太容易了！差點就要答應下來，直到被身邊的一位智者阻止。答案會變成 2 的 100 次方，也就是 $2676506 * 10$ 的 30 次方的驚人天文數字。挖空全國的稻米都不知道給不給得出來。

由此可知，**線性成長**、和指數型成長的速度差異，是非常巨大的！所以我們要尋求一種方法，來描述執行時間相對於問題大小的**成長速度**”。

這種方法，就是
規範出理論的上限值— 符號表達為「O」，唸作 Big-Oh

時間複雜度

三種搜尋法的時間複雜度比較

演算法	線性	二元.迭代	二元.遞迴
時間	2分42秒	26秒	9秒

時間複雜度

高斯解	土方法
<pre>1 sum = (1+100) * 100 / 2 2 print(sum)</pre>	<pre>1 sum = 0 2 for i in range(1, 101, 1): 3 sum = sum + i 4 print(sum)</pre>
$O(1)$	$O(n)$

空間複雜度

迴圈輸入	列表輸入
<pre>1 count = int(input()) 2 for i in range(0, count, 1): 3 num = int(input())</pre>	<pre>1 count = int(input()) 2 thislist = [] 3 for i in range(0, count, 1): 4 thislist.append(int(input()))</pre>
$O(1)$	$O(n)$