|  |  |  |  |
| --- | --- | --- | --- |
|  | Longer fisrt | Shorter first | Original |
| leftmost | 0.031 | 0.016 | 0.031 |
| median of three | 0.016 | 0.015 | 0.015 |

(單位:秒)

首先，先解釋tail recursion的對shorter first和longer first的不同影響。因為tail recursion optimization只有在function末端做遞迴呼叫時才會達到優化的效果。因此quick sort每次做的兩次遞迴呼叫，第一次仍然是使用系統堆疊，只有第二次才會被優化。要加快quick sort的速度，應該盡可能讓較多的部分使用tail recursion，所以，讓較短的sublist先行做遞迴，較長的sublist使用尾端遞迴才能達到最大優化。我們也可以看到shorter first的執行時間顯著地優於另外兩者。

而原本預期使用median of three可以大幅提升排序的效率，但效果卻沒有非常明顯的差別。推測可能是因為在決定三者中誰是pivot時，也額外花了一些時間在if/else的判斷。但若待排序的資料越來越大時，median of three的優勢應該會隨之增長。另外，我們也可以發現，在使用median of three時，left most/shorter first/original的影響就變的小很多。我認為是由於median of three這個步驟就已經減少了兩個sublists的不平衡情形，所以tail ecursion的優化能力就比較不會因為呼叫的順序有所區別。

輸出結果截圖:

Left-most / shorter first:



Left-most / longer first:



Left-most / original:



Median of three/original:



Median of three/longer first:



Median of three/shorter first:

