

# Heap Sort V.S. Merge Sort

05170229 巨資四B 袁嘉謙

	Heap Sort	Merge Sort
時間複雜度	$O(n \log n)$ sift down在將元素換位的過程中時間複雜度為 $O(\log n)$  排序時會進行 $n$ 次sift down的步驟  所以最終結果為 $O(n \log n)$	$O(n \log n)$ 將有 $n$ 個元素的陣列切分為單獨元素，共需 $(n-1)$ 次  兩兩合併需 $\log_2 n$ 個步驟且要花 $n$ 個回合完成 Ex. $4 > 2 > 1$  所以就是 $(n-1) + n * \log_2 n$
空間複雜度	$O(1)$ 不需額外創造空間	$O(n)$ 要創造與輸入值相同長度的空間來存放排序後的數列
個人想法	主要是函式較為複雜，在理解時容易卡關，會覺得好不容易做出了個MaxHeap但怎麼和預想的排列不一樣，覺得化為流程圖後會比較容易理解他的概念。	與最早所學的Quick Sort很類似，除了在為甚麼要兩兩切割在合併這點困惑了一下之外，感覺是一個比較平易近人一點點的演算法。

寫的過程有些忘記Heap Sort，參考資料: <https://rust-algo.club/sorting/heapsort/>