

How to compile my file:

Just type “make” 就可以 compile 出名為 merger 的執行檔

Test:

	N / 100	N / 25	N / 10	N / 5	N / 2
100	Real: 0.008s user : 0.002s sys: 0.010s	Real: 0.005s user : 0.002s sys: 0.004s	Real: 0.005s user : 0.001s sys: 0.003s	Real: 0.005s user : 0.001s sys: 0.002s	Real: 0.004s user : 0.001s sys: 0.002s
10000	Real: 0.10s user : 0.016s sys: 0.008s	Real: 0.11s user : 0.007s sys: 0.004s	Real: 0.09s user : 0.002s sys: 0.004s	Real: 0.07s user : 0.007s sys: 0.003s	Real: 0.08s user : 0.007s sys: 0.003s
1000000	Real: 0.79s user : 0.597s sys: 0.039s	Real: 0.82s user : 0.580s sys: 0.033s	Real: 0.84s user : 0.558s sys: 0.028s	Real: 0.85s user : 0.590s sys: 0.026s	Real: 0.91s user : 0.543s sys: 0.023s
10000000	Real: 7.84s user : 6.142s sys: 0.256s	Real: 8.08s user : 5.980s sys: 0.221s	Real: 8.57s user : 5.791s sys: 0.190s	Real: 8.53s user : 5.746s sys: 0.174s	Real: 8.19s user : 5.672s sys: 0.159s

觀察：

由上面的觀察發現，**real time** 會隨著每個 **segment** 的 **size** 不斷變大，而跟著變大（因為需要做更大的 **sort**，而 **sort** 的時間複雜度為 $N\log N$ ，所以整體時間會變慢）

然而，**user time** 會隨著需要合併的 **segment** 數減少及需要建的 **thread** 數減少而跟著減少（因為雖然 **merge** 只需 N 的時間複雜度，但如果其常數相當大時，其 **cost** 也會跟著變高，反之，就會變小）

在 **total DATA** 還小時，這樣的現象看不太出來，隨著 **total DATA** 不斷變大，這樣的現象就越來越明顯，尤其是在 $N=10000000$ 的時候最為明顯。