

Assignment 3 Compare Sort

- ✚ วัตถุประสงค์ เปรียบเทียบความเร็วในการเรียงลำดับของ quick sort , merge sort กับ qsort ที่อยู่ใน Library <stdlib.h>
- ✚ อ่านไฟล์ TEST.CSV ที่เตรียมไว้ให้ เก็บในอาร์เรย์
 - ข้อมูลในไฟล์เก็บบรรทัดละ 1 ตัว แบ่งข้อมูลเป็น 4 ฟิลด์ โดยมีเครื่องหมาย ',' comma คั่น
 - ข้อกำหนด ข้อมูลชุดแรกเป็นลำดับตัวนับ ชุดที่สองเป็นตัวเลขจำนวนเต็ม ชุดที่สามและสี่ให้เป็นสตริงความยาวไม่เกินชุดละ 64 ตัวอักษร
 - สร้างฟังก์ชัน Merge sort (number) เพื่อเรียงลำดับโดยใช้คีย์เป็นตัวเลขฟิลด์ที่สอง
 - สร้างฟังก์ชัน Merge sort (string) เพื่อเรียงลำดับโดยใช้คีย์เป็นตัวอักษรฟิลด์ที่สี่
 - สร้างฟังก์ชัน quick sort (number) เพื่อเรียงลำดับโดยใช้คีย์เป็นตัวเลขฟิลด์ที่สอง
 - สร้างฟังก์ชัน quick sort (string) เพื่อเรียงลำดับโดยใช้คีย์เป็นตัวอักษรฟิลด์ที่สี่
 - สร้าง compare function (number) เพื่อใช้กับ qsort โดยใช้คีย์เป็นตัวเลขฟิลด์ที่สอง
 - สร้าง compare function (string) เพื่อใช้กับ qsort โดยใช้คีย์เป็นตัวอักษรฟิลด์ที่สี่
 - จับเวลาเปรียบเทียบ
- ✚ คำแนะนำเพิ่มเติม
 1. ใช้โครงสร้างและฟังก์ชันเดิมใน Assignment 2 เพื่อจองตัวแปร และอ่านข้อมูล
 2. กำหนดให้เรียงลำดับข้อมูลทั้งหมดในไฟล์ TEST.CSV
 3. จองตัวแปร และสร้างอาร์เรย์ สำหรับเก็บข้อมูลที่อ่านได้ และ จองตัวแปรอีกชุดหนึ่ง สำหรับใช้ทดสอบการเรียงลำดับ(สร้างฟังก์ชัน copy ตัวแปร เพื่อไม่ต้องอ่านไฟล์ซ้ำ) merge sort , quick sort และ qsort
 4. ควรสร้างฟังก์ชันเพื่อทดสอบความถูกต้องของการเรียงลำดับที่เสร็จแล้วด้วย
 - Test case ให้แสดงผลข้อมูลตัวแรก(0) ตัวที่ (49999) และตัวสุดท้าย(99999) เพื่อตรวจสอบความถูกต้อง

Assignment 3 Test Case



Test Case

- แสดงผลข้อมูลตัวแรก data[0]
- ตัวกลาง data[49999]
- ตัวสุดท้ายที่อ่านได้ data[99999]

	number(ms)	String(ms)
library sort		
Quick sort		
Merge sort		

test.csv	order	numer	string1	string2
unsort data[0]	0	111111111111	AAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAA	ZZZZZZZZZZZZZZZZZZZZZZZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZ
data[49999]	49999	725622549999	EAHCDMRGCOHEVVCTJGFSDXZMZU SQTZFZIEHQCCWWUVMYMKKKKKK	EQEVXBWZOYLTJAWDEILNMTRPCH XHGOIWCEEFBQJAXKSAIKKKKKK
data[99999]	99999	99999990000	ZZZZZZZZZZZZZZZZZZZZZZZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZ	AAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAA

sort number data[0]	46581	10000000000	ABCDEFGHIJKLMNPOQRSTUVWXYZ ABCDEFGHIJKLMNPOQRSTUVWXYZ	ABCDEFGHIJKLMNPOQRSTUVWXYZ ABCDEFGHIJKLMNPOQRSTUVWXYZ
data[49999]	43974	55652616000	RINZGHRJWITFMUHTNNNXIFMVNG GOYIDLBNUIIJHEVQEBNZAXMMM	YCDKDIPVNIHDHDKORMPQEGKAI RVANUZNPIIJDHZMIDGMPDMMM
data[99999]	29423	99999999999	ZYXWVUTSRQPONMLKJIHGFEDCBA ZYXWVUTSRQPONMLKJIHGFEDCBA	ZYXWVUTSRQPONMLKJIHGFEDCBA ZYXWVUTSRQPONMLKJIHGFEDCBA

sort string data[0]	99999	99999990000	ZZZZZZZZZZZZZZZZZZZZZZZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZ	AAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAA
data[49999]	76419	386637875877	PBUNVJEALEITEFHGFPEPGWMMYA WHOUKDFRTCOLXFIZUZALYCTPXR	NAQTJKVVGLEBEBXULCUNUVLJNM AWOQNQBMASEBZLQKIWLXLCUIGZX
data[99999]	0	111111111111	AAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAA	ZZZZZZZZZZZZZZZZZZZZZZZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZ