11027149資訊三甲游婕歆

1. 開發環境

C++, 在windows上wsl用ubuntu編譯執行

1. 實作方法及流程

第1題沒有用到process單純把整份資料進行氣泡排序

第2題則是在method2這個function先把資料切成k份，再做單一process，先做k份資料氣泡排序再做k-1次合併。

第3題是在method3這個function先把資料切成k份，透過迴圈配合fork建立k個process執行氣泡排序，再建立k-1個process實作合併，透過共享記憶體讓子process處理的資料能讓父process收到。

第4題是在method4這個function先把資料切成k份，建立thread先透過vector建立k個執行緒，再透過迴圈一個一個跑氣泡排序，接著是在k-1個執行緒下merge sort。

Merge()實作方法:每次迴圈都會將k份中的前兩份數據合併為一份，直到k份中只剩下一份數據，也就是所有的數據都合併為一份。

因為用了c++來寫不像python有函式庫可以直接import來用，所以方法3我做了很久，本來是用pipe方式來做，但父process一直都接收不到子process做完的資料，我都想把整個程式碼改成python，但我就一直上網查還有問chatgpt，才找到共享記憶裡的方式，讓它們從同一個地方存取資料，才不會造成最後甚麼都印不出來。

1. 探討結果與原因

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K={10,100} | N=1w | N=10w | N=50w | N=100w |
| 方法1 | 0.467995 s,  0.469595 s | 54.3422 s,  54.22 s | 11542.86s  11492.5s | 32711.63s  34081.23s |
| 方法2 | 0.0302584 s  0.01024 s | 4.64393 s,  0.43373 s | 123.259 s,  12.4988 s | 618.24 s,  248.54 s |
| 方法3 | 0.0116884 s  0.0184 s | 1.48268 s,  0.170961 s | 40.4255 s,  4.04023 s | 106.965 s,  15.3349s |
| 方法4 | 0.0466858 s  0.048 s | 1.55885 s,  0.179084 s | 43.9287 s,  4.03363 s | 188.583 s,  15.1425 s |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N={10W,50W} | K=10 | K=100 | K=500 | K=1000 |
| 方法1 | 3.467995 s  11542.86s | 3.46295 s  10242.46s | 3.4395 s  10223.44s | 3.3499 s  10231.339s |
| 方法2 | 2.64393 s  123.259 s | 0.43373 s  12.4988 s | 0.08829 s  1.1277 s | 0.32125 s  3.15165 s |
| 方法3 | 1.48268 s  40.4255 s | 0.170961 s  4.04023 s | 0.117338 s  0.489251 s | 0.232134 s  1.20912 s |
| 方法4 | 1.55885 s  43.9287 s | 0.179084 s  4.03363 s | 0.129008 s  0.542835 s | 0.14024 s  4.48395 s |

由上面表格可看出相同方法中K值越大資料被切得多份，執行時間越短，效率越好可能是因為將區塊變小降低了計算的複雜度也可以減少規劃的空間。而4種方法的比較 : 單純只有bubblesort的方法1會是最慢的效率最差，其他的方法時間雖然有差 但是因為方法一時間太大在折線圖中看不出明顯差異，所以我另外做了一張圖不包含方法1，就能看出方法2、3、4的差異，可以看出在處理同一份資料時，方法2是最慢的，而3、4的時間是差不多的，因為方法2士單process一次只能做一件事所以效率會比其他兩個法差一點。