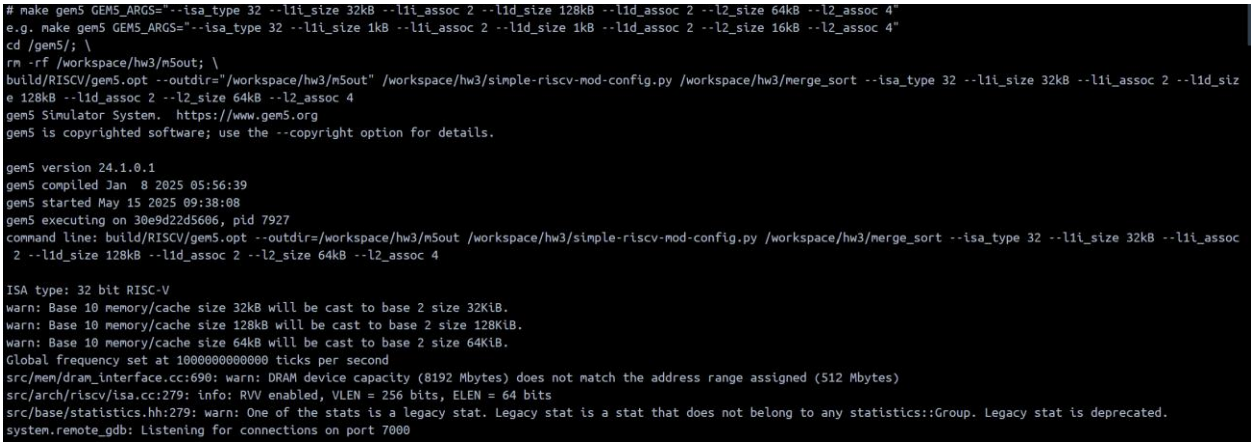
Computer Architecture - Homework 3 Report

StudentID:B10505050 Name:張鈞民

Part1. Add new components to gem5 config.

- **1-1: Implement L2 cache in gem5 config**
- **1-2: Draw and analyze the program results based on the config**

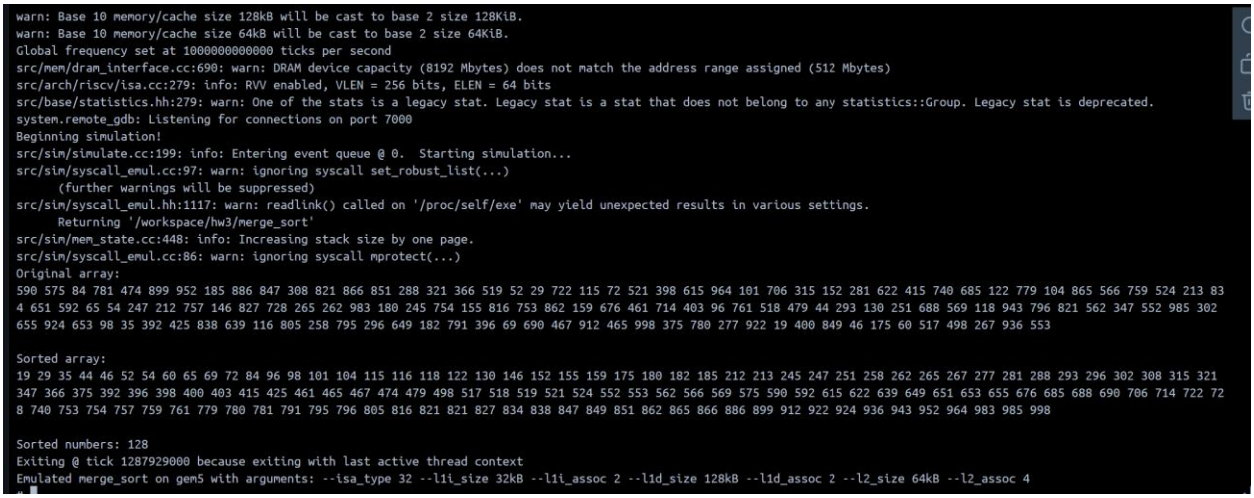
Part1-1. Show screenshots of your program running successfully (refer to P.11) (15%)



```
# make gem5 GEM5_ARGS="--isa_type 32 --l1i_size 32kB --l1i_assoc 2 --l1d_size 128kB --l1d_assoc 2 --l2_size 64kB --l2_assoc 4"
e.g. make gem5 GEM5_ARGS="--isa_type 32 --l1i_size 1kB --l1i_assoc 2 --l1d_size 1kB --l1d_assoc 2 --l2_size 16kB --l2_assoc 4"
cd /gem5/; \
rm -rf /workspace/hw3/m5out; \
build/RISCV/gem5.opt --outdir="/workspace/hw3/m5out" /workspace/hw3/simple-riscv-mod-config.py /workspace/hw3/merge_sort --isa_type 32 --l1i_size 32kB --l1i_assoc 2 --l1d_siz
e 128kB --l1d_assoc 2 --l2_size 64kB --l2_assoc 4
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version 24.1.0.1
gem5 compiled Jan  8 2025 05:56:39
gem5 started May 15 2025 09:38:08
gem5 executing on 30e9d22d5606, pid 7927
command line: build/RISCV/gem5.opt --outdir=/workspace/hw3/m5out /workspace/hw3/simple-riscv-mod-config.py /workspace/hw3/merge_sort --isa_type 32 --l1i_size 32kB --l1i_assoc
 2 --l1d_size 128kB --l1d_assoc 2 --l2_size 64kB --l2_assoc 4

ISA type: 32 bit RISC-V
warn: Base 10 memory/cache size 32kB will be cast to base 2 size 32KiB.
warn: Base 10 memory/cache size 128kB will be cast to base 2 size 128KiB.
warn: Base 10 memory/cache size 64kB will be cast to base 2 size 64KiB.
Global frequency set at 1000000000000 ticks per second
src/mem/dram_interface.cc:690: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
src/arch/riscv/isa.cc:279: info: RVV enabled, VLEN = 256 bits, ELEN = 64 bits
src/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is a stat that does not belong to any statistics::Group. Legacy stat is deprecated.
system.remote_gdb: Listening for connections on port 7000
```



```
warn: Base 10 memory/cache size 128kB will be cast to base 2 size 128KiB.
warn: Base 10 memory/cache size 64kB will be cast to base 2 size 64KiB.
Global frequency set at 1000000000000 ticks per second
src/mem/dram_interface.cc:690: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
src/arch/riscv/isa.cc:279: info: RVV enabled, VLEN = 256 bits, ELEN = 64 bits
src/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is a stat that does not belong to any statistics::Group. Legacy stat is deprecated.
system.remote_gdb: Listening for connections on port 7000
Beginning simulation!
src/sim/simulate.cc:199: info: Entering event queue @ 0.  Starting simulation...
src/sim/syscall_emul.cc:97: warn: ignoring syscall set_robust_list(...)
      (further warnings will be suppressed)
src/sim/syscall_emul.hh:1117: warn: readlink() called on '/proc/self/exe' may yield unexpected results in various settings.
      Returning '/workspace/hw3/merge_sort'
src/sim/mem_state.cc:448: info: Increasing stack size by one page.
src/sim/syscall_emul.cc:86: warn: ignoring syscall mprotect(...)
Original array:
590 575 84 781 474 899 952 185 886 847 308 821 866 851 288 321 366 519 52 29 722 115 72 521 398 615 964 101 706 315 152 281 622 415 740 685 122 779 104 865 566 759 524 213 83
4 651 592 65 54 247 212 757 146 827 728 265 262 983 180 245 754 155 816 753 862 159 676 461 714 403 96 761 518 479 44 293 130 251 688 569 118 943 796 821 562 347 552 985 302
655 924 653 98 35 392 425 838 639 116 805 258 795 296 649 182 791 396 69 690 467 912 465 998 375 780 277 922 19 400 849 46 175 60 517 498 267 936 553

Sorted array:
19 29 35 44 46 52 54 60 65 69 72 84 96 98 101 104 115 116 118 122 130 146 152 155 159 175 180 182 185 212 213 245 247 251 258 262 265 267 277 281 288 293 296 302 308 315 321
347 366 375 392 396 398 400 403 415 425 461 465 467 474 479 498 517 518 519 521 524 552 553 562 566 569 575 590 592 615 622 639 649 651 653 655 676 685 688 690 706 714 722 72
8 740 753 754 757 759 761 779 780 781 791 795 796 805 816 821 821 827 834 838 847 849 851 862 865 866 886 899 912 922 924 936 943 952 964 983 985 998

Sorted numbers: 128
Exiting @ tick 1287929000 because exiting with last active thread context
Emulated merge_sort on gem5 with arguments: --isa_type 32 --l1i_size 32kB --l1i_assoc 2 --l1d_size 128kB --l1d_assoc 2 --l2_size 64kB --l2_assoc 4
```

Part1-1. Show screenshots of your program and cache summary (refer to P.12) (15%)

<program summary>

```
Program summary
----------------------------------------
simulated time            | 0.001288 s
simulated tick            | 1,287,929,000 ticks
total Inst.               | 378,971 instructions
total cycle               | 1,287,929 cycles
CPI                       | 3.396007
IPC                       | 0.294463
Int-Inst. count           | 376,040 instructions
Load-Inst. count          | 80,758 instructions
Store-Inst. count         | 40,786 instructions
Vector-Inst. count        | 0 instructions
```

<L1 cache summary>

```
L1-Instruction-Cache summary
-------------------------------------
$L1-I hit count           | 459,287 counts
$L1-I miss count          | 323 counts
$L1-I access count        | 459,610 counts
$L1-I miss rate           | 0.07% miss rate
L1-I assoc                | 2
L1-I size                 | 32768

L1-Data-Cache summary
-------------------------------------
$L1-D hit count           | 120,577 counts
$L1-D miss count          | 929 counts
$L1-D access count        | 121,506 counts
$L1-D miss rate           | 0.76% miss rate
L1-D assoc                | 2
```

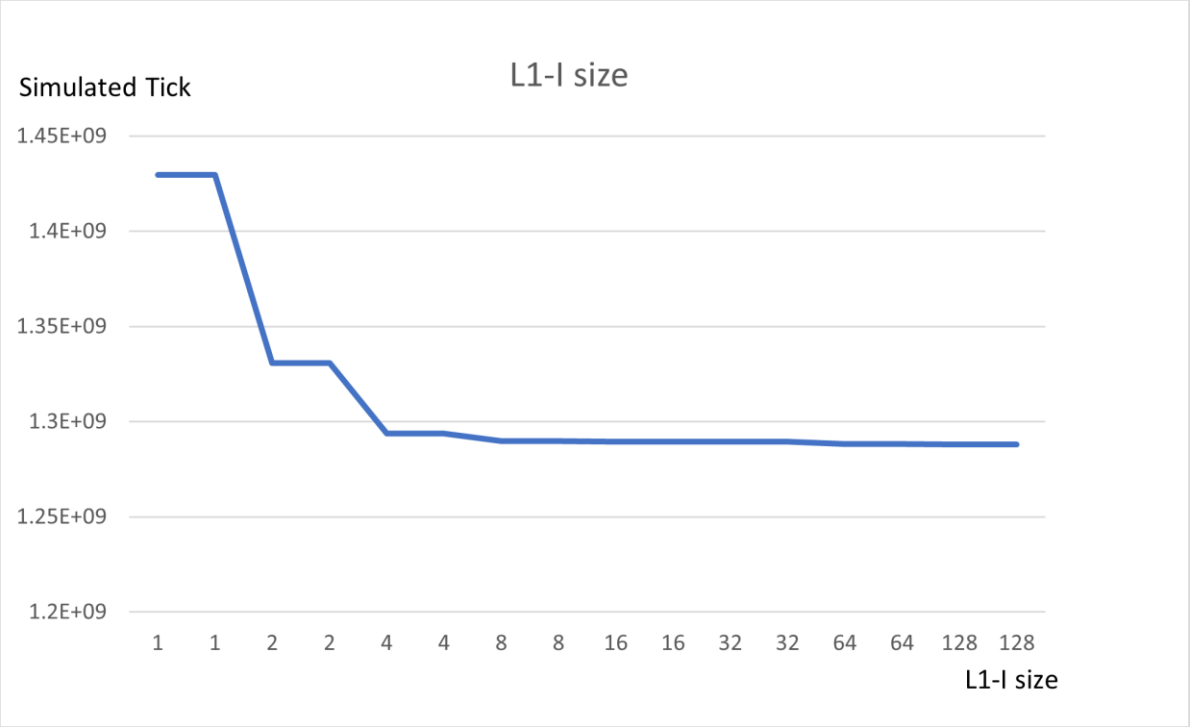
```
L1-Data-Cache summary
-------------------------------------
$L1-D hit count           | 120,577 counts
$L1-D miss count          | 929 counts
$L1-D access count        | 121,506 counts
$L1-D miss rate           | 0.76% miss rate
L1-D assoc                | 2
L1-D size                 | 131072
```

<L2 cache summary>

```
L2-Cache summary
-------------------------------------
$L2 hit count             | 7 counts
$L2 miss count            | 1,246 counts
$L2 access count          | 1,253 counts
$L2 miss rate             | 99.44% miss rate
L2 assoc                  | 4
L2 size                   | 65536
```

Part1-2. Draw a graph based on the different hardware config (refer to P.13-14). (40%)

<L1-I cache size>



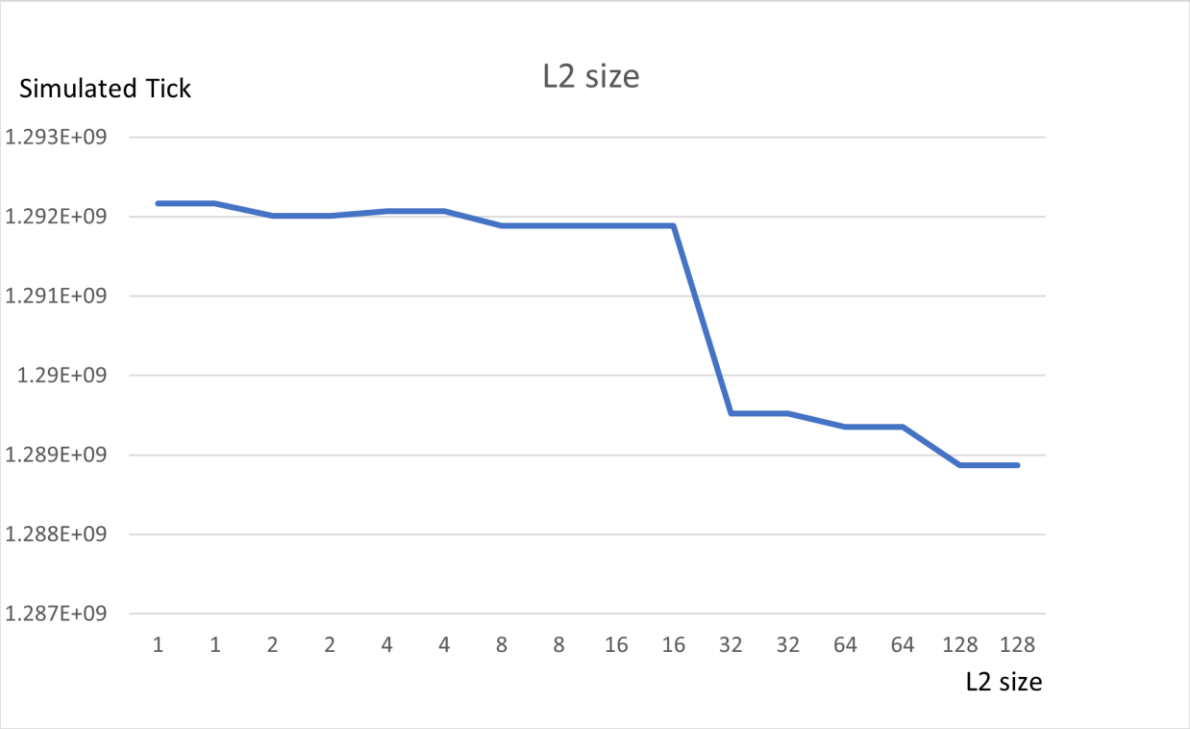
Other Variable :

ll1i_assoc 2 --l1d_size 32kB --l1d_assoc 2 --l2_size 64kB --l2_assoc 4

<L1-D cache size>

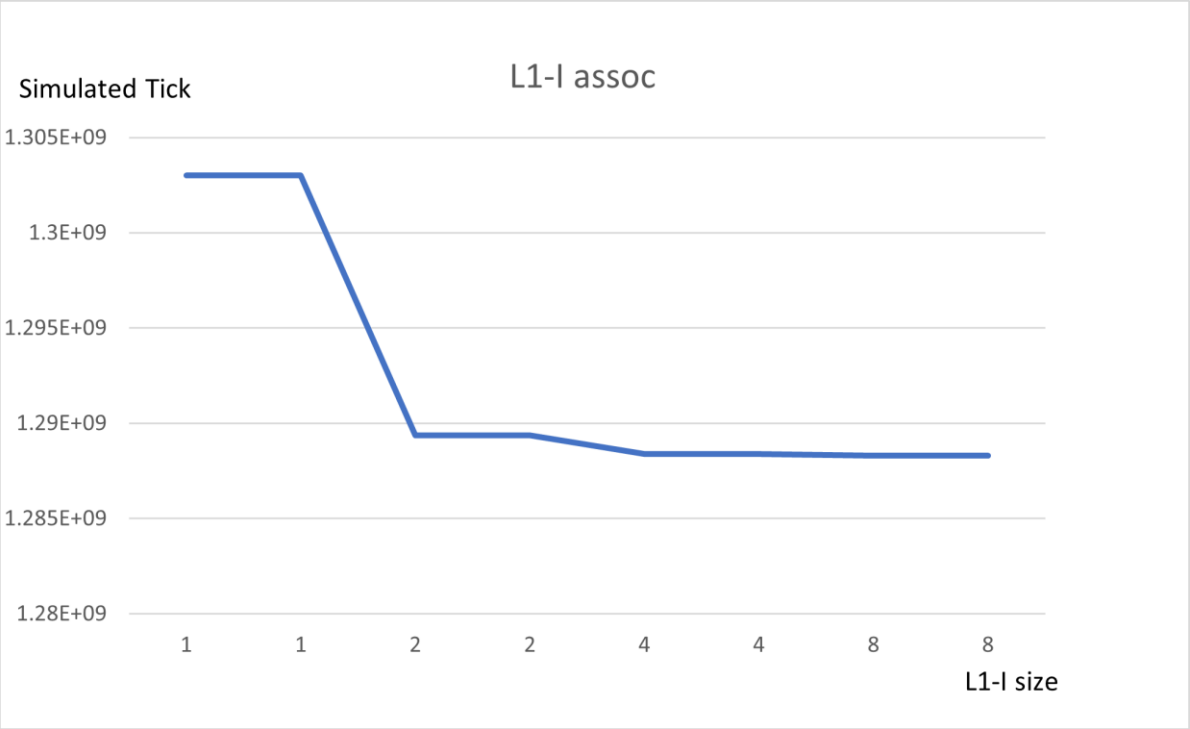
Other Variable :
l1i_size 32kB --l1i_assoc 2 --l1d_assoc 2 --l2_size 64kB --l2_assoc 4

<L2 cache size>



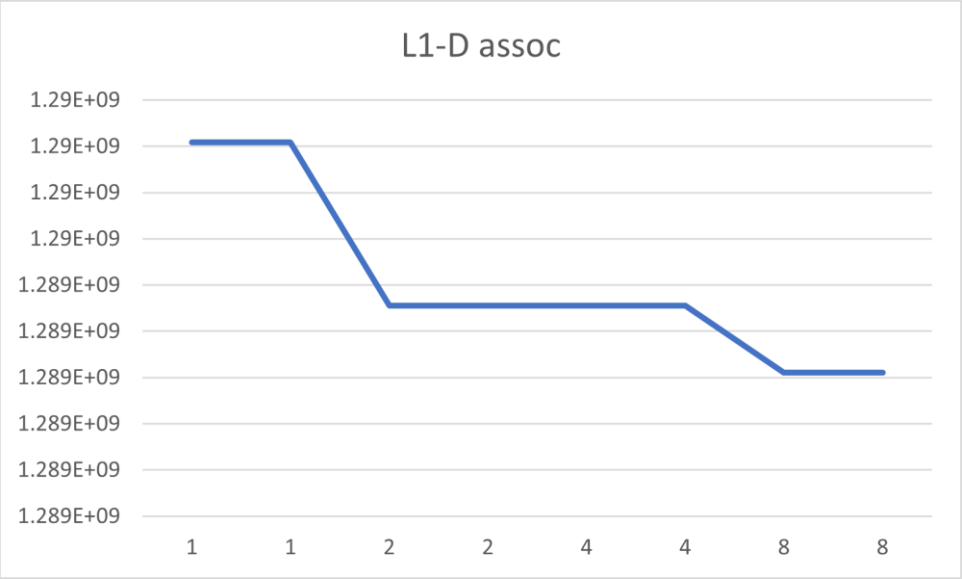
Other Variable :
l1i_size 32kB --l1i_assoc 2 --l1d_size 32kB --l1d_assoc 2 --l2_assoc 4

<L1-I associativity>



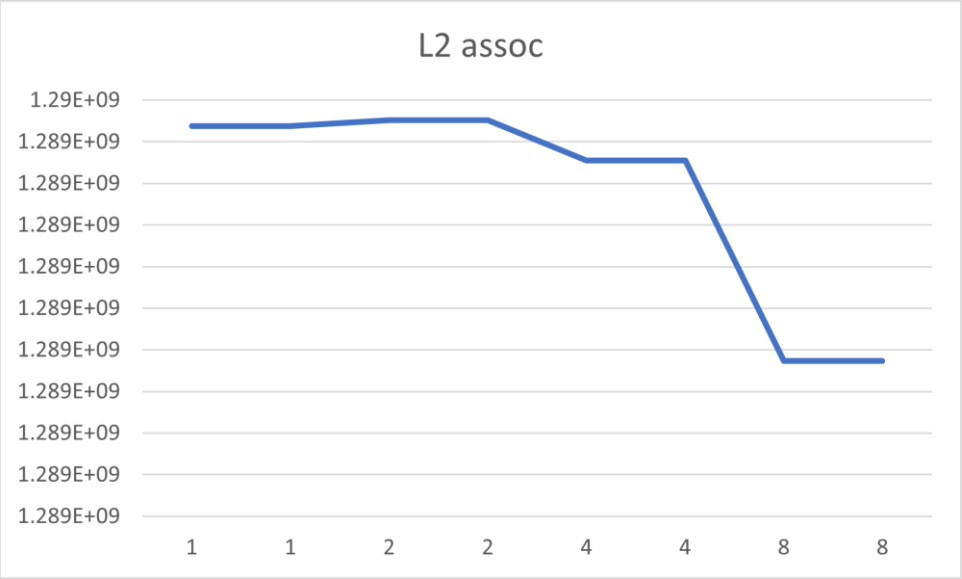
Other Variable :
l1i_size 32kB --l1d_size 32kB --l1d_assoc 2 --l2_size 64kB --l2_assoc 4

<L1-D associativity>



Other Variable :
l1i_size 32kB --l1i_assoc 2 --l1d_size 32kB --l2_size 64kB --l2_assoc 4"

<L2 associativity>



Other Variable :
l1i_size 32kB --l1i_assoc 2 --l1d_size 32kB --l1d_assoc 2 --l2_size 64kB

Part2. Analyze and find the optimal config based on the application

- **Cache Size: 1KB ~ 128KB**
- **Associativity: 1 ~ 8**

Part2-1. Find the optimal config settings for this program. (10%)

Config	Cache size (KB)	Associativity
L1-I cache	32	8
L1-D cache	64	8
L2 cache	1	8

Performance	Score
1,287,574,000	23-11 = 12

※ Save your optimal config in gem5_args.conf

Part2-2. Explain why this config achieves optimal performance ? (20%)

一開始先從已知的數值內找出最好的解如下

Ticks = 1,287,929,000

L1-I: 32kB 2-way

L1-D: 128kB 2-way

L2 ：64kB 4-way

由於 associativity 在某些情況增高可以降低 ticks 數，且也不會影響 score，故先從 associativity 下手，但如前所述，只有在某些情況會降低，有時候調高反而會增加 ticks(因為要比對 tag)

故在調整 assoc 的同時也要注意該 cache 的大小，最後調整完後，結果為以下

Ticks : 1,287,574,000

L1-I: 32kB 8-way

L1-D: 64kB 8-way

L2 ：1kB 8-way

分析原因：

1 . 高度的 associativity，由於 merge sort 會一直訪問資料(from recursive)，故若 cache 的 hit rate 太低，造成多次的 miss，就會增加 ticks，所以選擇將 associativity 在可以接受的(使降低而非增加)的 cache 大小內增至最大。

2 .調整 L1 I 使 cache 足以儲存整個排序程式的指令，降低 instruction miss。

3 .由於排序中會要存放多段 array，故 L1-D 應該有一個底線的容量，在這個前提下，得到的最好解是 64，當降低容量，並依照情況調整 associativity 時，ticks 只要小於 64，時間幾乎都會暴增到 129 的 scale，因此選用 64

4. 剩下的 L2，同樣在能維持高 associativity 的情況下盡量降低容量，使後續 score 能降低影響，所以盡可能的壓低容量