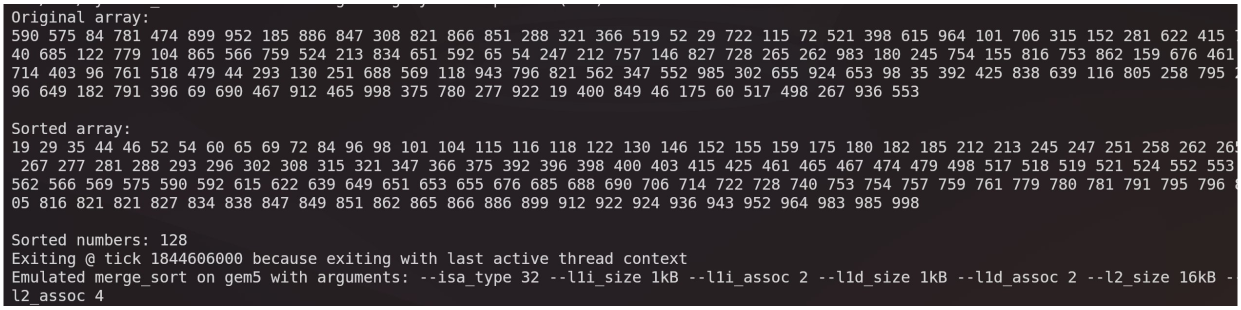
Computer Architecture - Homework 3 Report

StudentID:B11901164 Name:陳秉緯

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%)



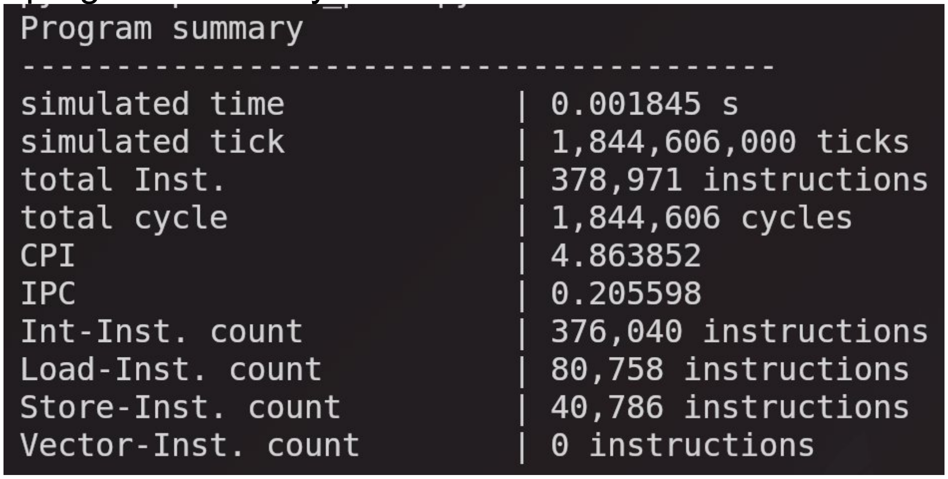
```
Original array:
590 575 84 781 474 899 952 185 886 847 308 821 866 851 288 321 366 519 52 29 722 115 72 521 398 615 964 101 706 315 152 281 622 415
40 685 122 779 104 865 566 759 524 213 834 651 592 65 54 247 212 757 146 827 728 265 262 983 180 245 754 155 816 753 862 159 676 461
714 403 96 761 518 479 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
96 649 182 791 396 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 26
 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
562 566 569 575 590 592 615 622 639 649 651 653 655 676 685 688 690 706 714 722 728 740 753 754 757 759 761 779 780 781 791 795 796
05 816 821 821 827 834 838 847 849 851 862 865 866 886 899 912 922 924 936 943 952 964 983 985 998

Sorted numbers: 128
Exiting @ tick 1844606000 because exiting with last active thread context
Emulated merge_sort on gem5 with arguments: --isa_type 32 --l1i_size 1kB --l1i_assoc 2 --l1d_size 1kB --l1d_assoc 2 --l2_size 16kB -
l2_assoc 4
```

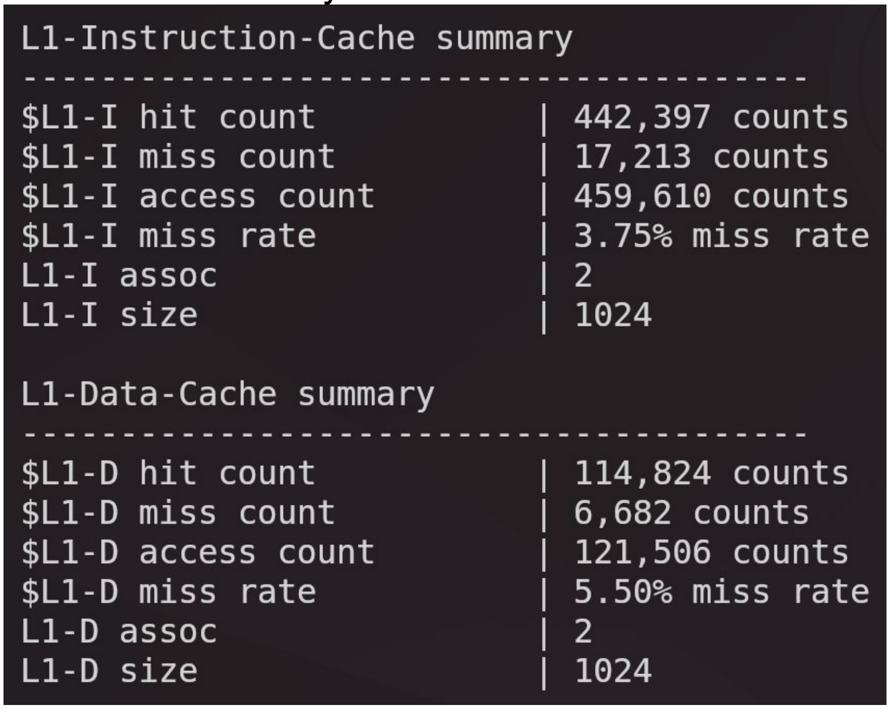
Part1-1. Show screenshots of your program and cache summary (refer to P.12) (15%)

<program summary>



```
Program summary
----------------------------------------
simulated time              | 0.001845 s
simulated tick              | 1,844,606,000 ticks
total Inst.                 | 378,971 instructions
total cycle                 | 1,844,606 cycles
CPI                         | 4.863852
IPC                         | 0.205598
Int-Inst. count             | 376,040 instructions
Load-Inst. count            | 80,758 instructions
Store-Inst. count           | 40,786 instructions
Vector-Inst. count          | 0 instructions
```

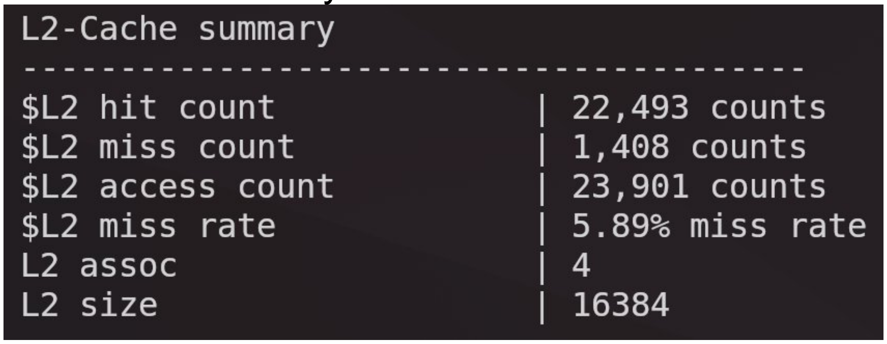
<L1 cache summary>



```
L1-Instruction-Cache summary
----------------------------------------
$L1-I hit count              | 442,397 counts
$L1-I miss count             | 17,213 counts
$L1-I access count           | 459,610 counts
$L1-I miss rate              | 3.75% miss rate
L1-I assoc                   | 2
L1-I size                    | 1024

L1-Data-Cache summary
----------------------------------------
$L1-D hit count              | 114,824 counts
$L1-D miss count             | 6,682 counts
$L1-D access count           | 121,506 counts
$L1-D miss rate              | 5.50% miss rate
L1-D assoc                   | 2
L1-D size                    | 1024
```

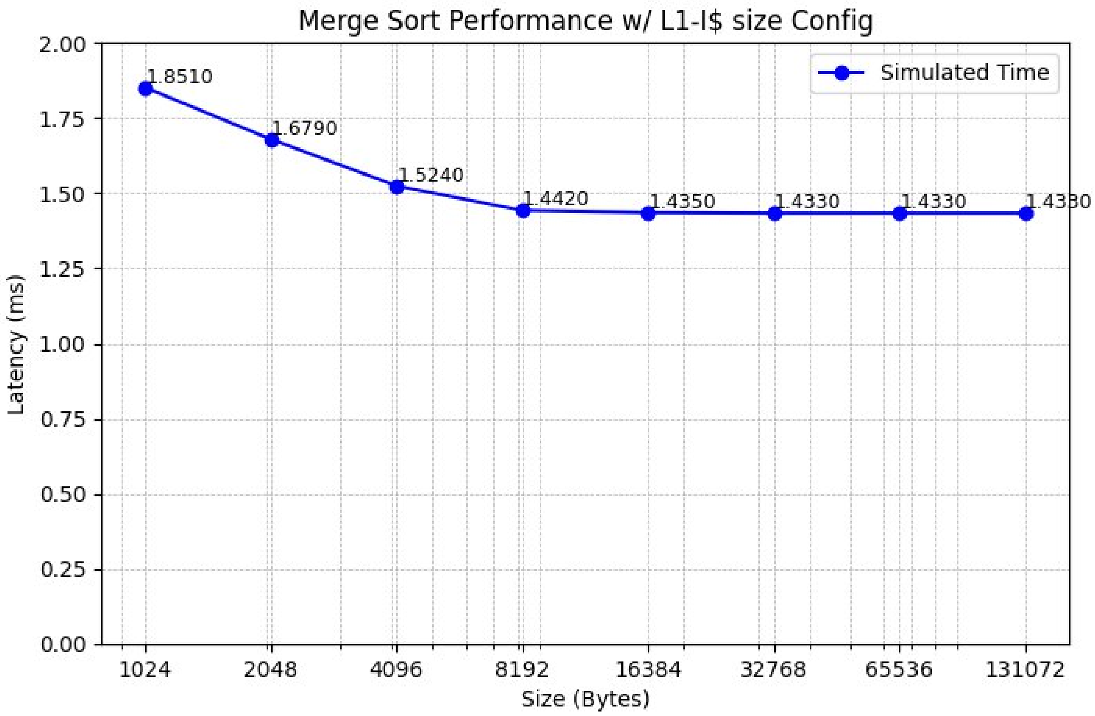
<L2 cache summary>



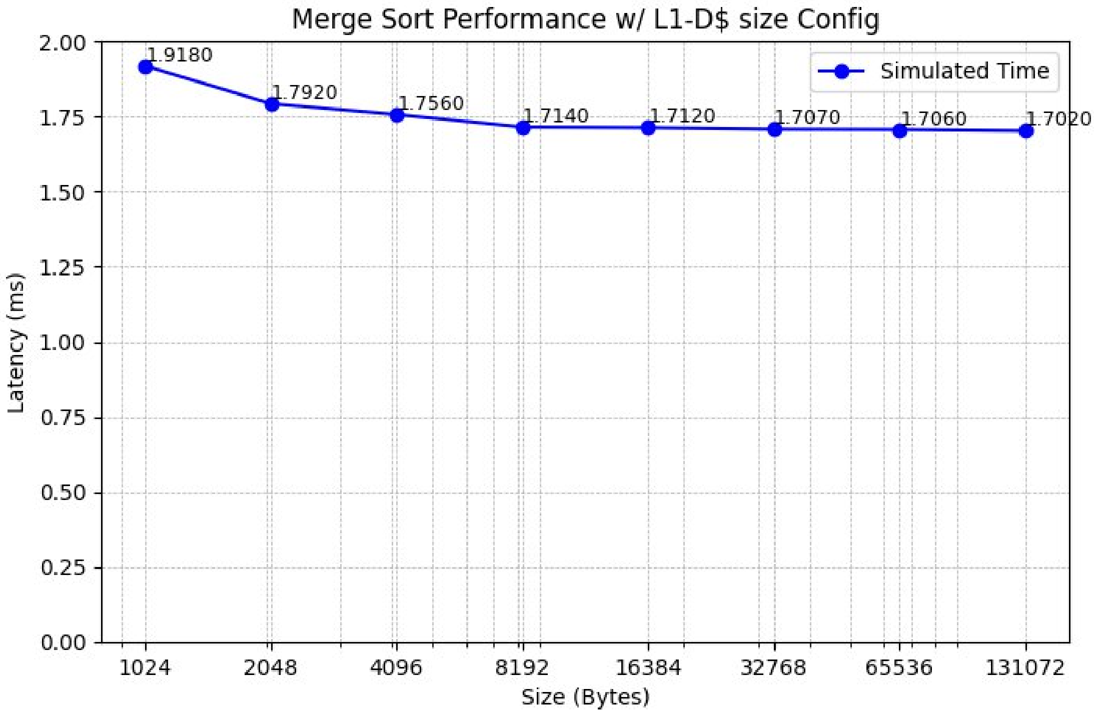
```
L2-Cache summary
----------------------------------------
$L2 hit count                | 22,493 counts
$L2 miss count               | 1,408 counts
$L2 access count             | 23,901 counts
$L2 miss rate                | 5.89% miss rate
L2 assoc                     | 4
L2 size                      | 16384
```

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

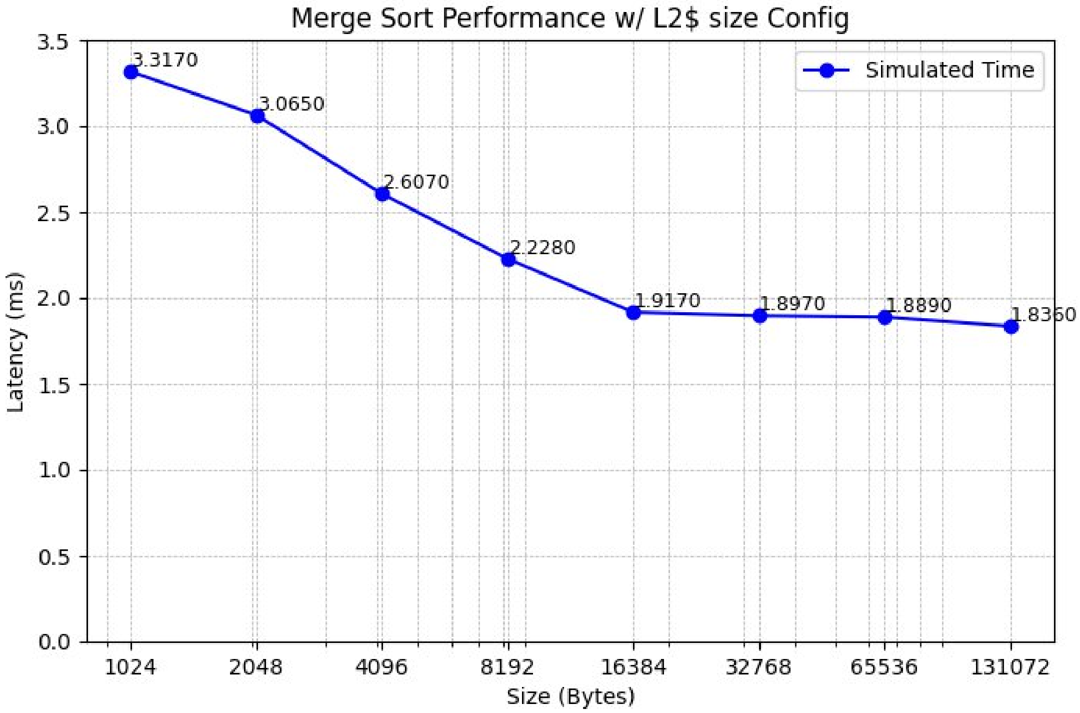
<L1-I cache size>


Merge Sort Performance w/ L1-I$ size Config
Simulated Time
Latency (ms)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.8510
1.6790
1.5240
1.4420
1.4350
1.4330
1.4330
1.4330
1024
2048
4096
8192
16384
32768
65536
131072
Size (Bytes)

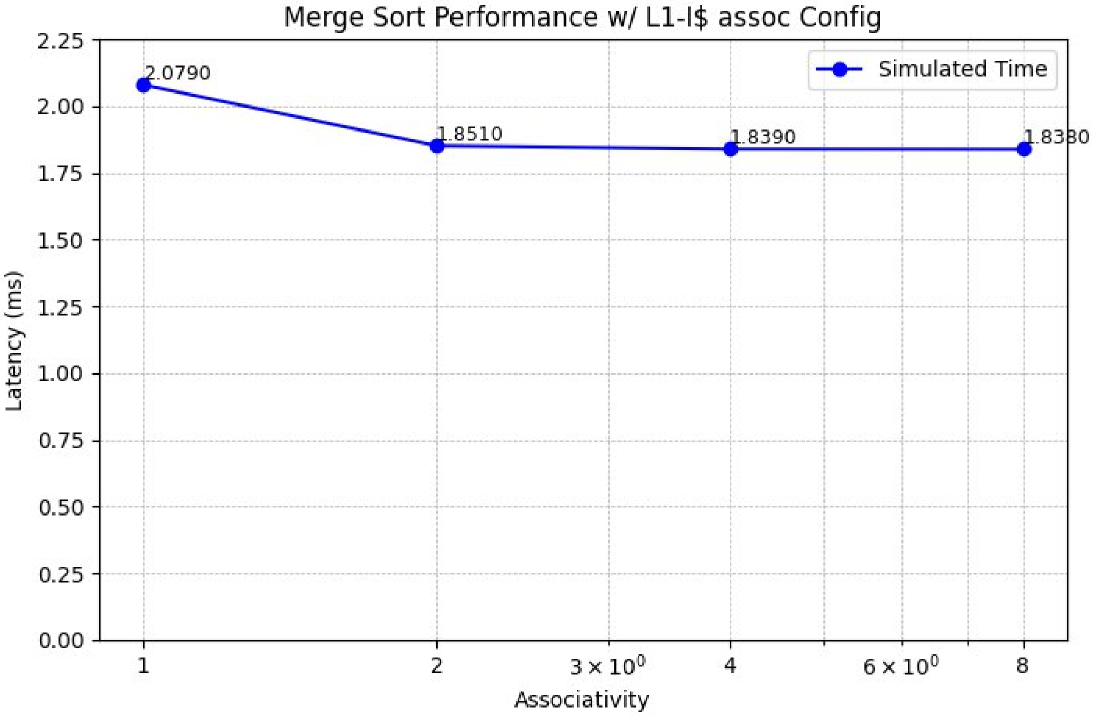
<L1-D cache size>


Merge Sort Performance w/ L1-D$ size Config
Simulated Time
Latency (ms)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1.9180
1.7920
1.7560
1.7140
1.7120
1.7070
1.7060
1.7020
1024
2048
4096
8192
16384
32768
65536
131072
Size (Bytes)

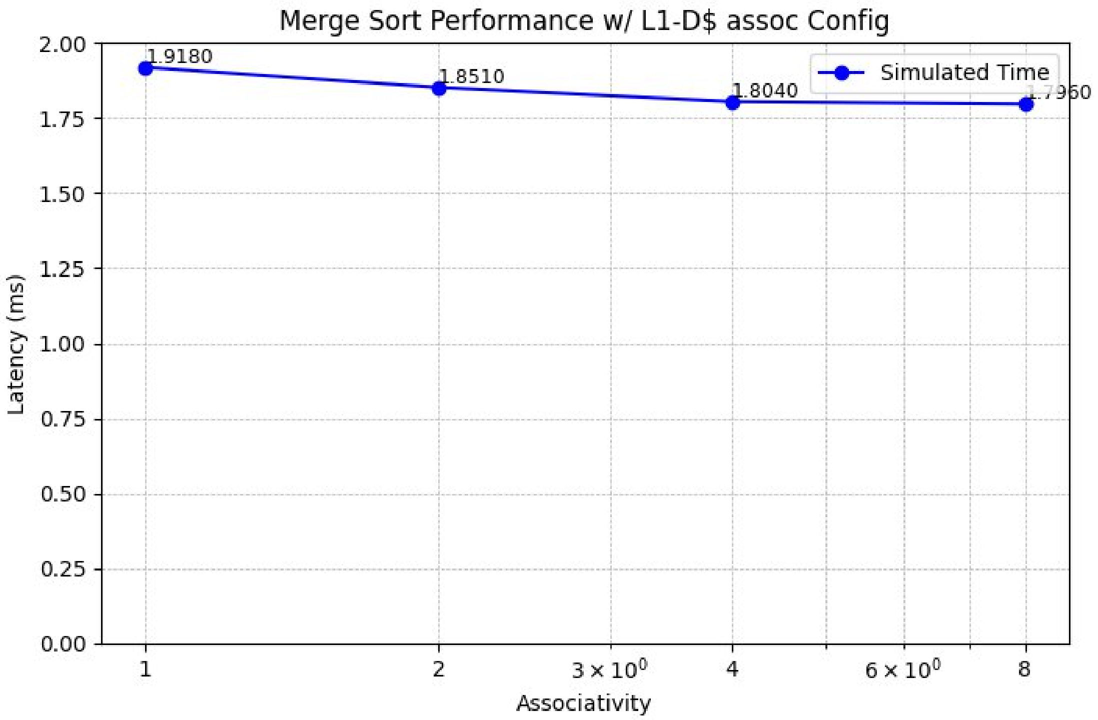
<L2 cache size>


Merge Sort Performance w/ L2$ size Config
Simulated Time
Latency (ms)
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
3.3170
3.0650
2.6070
2.2280
1.9170
1.8970
1.8890
1.8360
1024
2048
4096
8192
16384
32768
65536
131072
Size (Bytes)

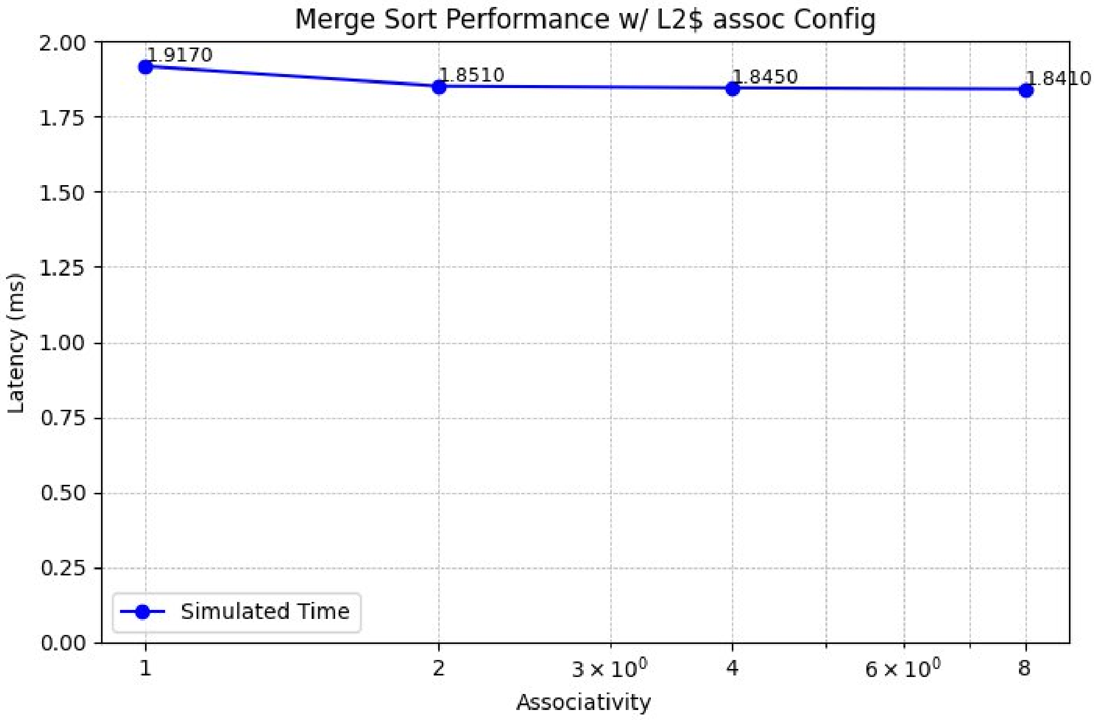
<L1-I associativity>


Merge Sort Performance w/ L1-I$ assoc Config
Simulated Time
Latency (ms)
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
2.0790
1.8510
1.8390
1.8380
1
2
3×10^0
4
6×10^0
8
Associativity

<L1-D associativity>



<L2 associativity>



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
1287574000	12

※ Save your optimal config in gem5_args.conf

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

1. L1-I Cache and L1-D Cache size and associativity:
   - These sizes are large enough to accommodate most of the frequently accessed instructions and data.
   - The high associativity (8-way) reduces conflict misses, allowing the cache to store data more effectively.
2. L2 Cache size and associativity:
   - The optimal L1 cache sizes are already sufficient to hold most data, leading to fewer L2 accesses.
   - The high associativity (8-way) reduces conflict misses, allowing the cache to store data more effectively.