# Test #1

截止日期 5月5日 23:59
分數 100
問題 40
可用 5月1日 0:01 - 5月5日 23:59
時間限制 60 分鐘

# 說明

**Canvas calls this a "Quiz", but it is really Test #1.**

**It consists of 40 multiple choice questions to be done in 60 minutes. It is Open Notes.**

**Once you start, you must finish. Canvas will not let you pause and come back.**

## 嘗試記錄

| | 嘗試 | 時間 | 分數 |
|---|---|---|---|
| **最新的** | 嘗試 1 | 43 分鐘 | 得分：100；總分：100 |

ⓘ 正確答案將於 5月6日 0:01 可用。

此測驗的分數： 得分：100；總分：100
已提交5月3日 12:15
此嘗試持續 43 分鐘。

問題 1
2.5 / 2.5 分數

**Our class's "Inverse Amdahl's Law" that you used in Projects #0 and #1 computes:**

- (●) Fp, given Sn and n
- ( ) Sn, given Fp and n
- ( ) Thread Efficiency, given Sn and n
- ( ) n, given Sn and Fp

問題 2

2.5 / 2.5 分數

## In terms of 8-byte double-precision numbers, the size of a cache line is:

- (●) 8 double-precision numbers
- ( ) 4 double-precision numbers
- ( ) 16 double-precision numbers
- ( ) 32 double-precision numbers

問題 3

2.5 / 2.5 分數

## The difference between static and dynamic scheduling of an OpenMP for-loop is:

- ( ) Dynamic scheduling divides all the for-loop passes among the threads at first
- (●) Dynamic scheduling divides only some of the for-loop passes among the threads at first
- ( ) Dynamic scheduling allows you to change how the for-loop passes are divided up while they are running
- ( ) Dynamic scheduling changes the chunksize while the for-loop is running

問題 4

2.5 / 2.5 分數

## The two types of coherence that caches want to see in order to deliver maximum performance are:

- (●) Spatial and Temporal
- ( ) Spatial and Thermal
- ( ) Systemic and Temporal
- ( ) Systemic and Thermal

問題 5

2.5 / 2.5 分數

## A Deadlock condition is when:

- (●) Two threads are each waiting for the other one to do something
- ( ) The CPU chip cannot find any more instructions to execute while waiting for a memory fetch
- ( ) When you keep internal state
- ( ) When it is a race to see which of two threads get to a piece of code first

問題 6

2.5 / 2.5 分數

## Hyperthreading is:

- Adding extra cache space
- Keeping one or more extra thread states within a core
- Adding more memory bandwidth
- Adding one or more cores

問題 7

2.5 / 2.5 分數

## A good way to make a piece of code *not* Thread Safe is to:

- Use a mutual exclusion lock
- Use a private variable
- Keep internal state
- Use a chunksize of 1

問題 8

2.5 / 2.5 分數

## A Private variable differs from a Shared variable in that:

- When each thread writes to it, the value goes to the same memory address
- Each thread has its own copy of it
- Writing to it automatically triggers a cache line reload
- Writing to it automatically triggers a power-of-two reduction operation

問題 9

2.5 / 2.5 分數

## OpenMP Reductions are faster than Atomic or Critical because:

- They sum into an array whose elements are a Fibonacci series in size
- They momentarily disable interrupts to keep the summing equation from being corrupted
- They sum into a user-supplied array and then let the programmer decide how to best sum them
- They sum into a separate variable per thread and then perform power-of-two addition

問題 10

2.5 / 2.5 分數

## To get an A in CS 475/575 requires:

- A weighted average of 93%
- 1060 total points
- A weighted average of 96%

問題 11

2.5 / 2.5 分數

## The word “deterministic” means:

- The same inputs will always produce the same outputs
- The program outputs change whenever you change the number of threads
- It describes a quantity that you are attempting to determine
- The program outputs change every time you run the program

問題 12

2.5 / 2.5 分數

## Why is there a photo of a carton of eggs in the Cache notes?

- It explains Temporary Coherence
- Bringing home a dozen eggs when you only need 2 today is like the way cache works
- Because the size of a cache line is a dozen floats
- It explains Stationary Coherence

問題 13

2.5 / 2.5 分數

## When multiplying two arrays together, you've decided you want to do it *4*-pairs-of-numbers at a time. The type of SIMD you should use is:

- SSE
- AVX
- AVX-512
- MMX

問題 14

2.5 / 2.5 分數

## A “race condition” is one where:

- (●) You get a different result depending on which thread gets to a piece of code first
- ( ) You get the same result regardless of which thread gets to a piece of code first
- ( ) It matters which stack holds a particular variable
- ( ) It matters which thread gets to a barrier first

問題 15

2.5 / 2.5 分數

## In an n-core multicore program, what do you need to do to compute the $F_{parallel}$?

- ( ) Go find out the size of the cache and use the inverse Amdahl's Law
- ( ) Figure out how many CPU sockets are in use and use the inverse Amdahl's Law
- ( ) Measure just the 20-core performance and use the inverse Amdahl's Law
- (●) Measure the Speedup and use our inverse Amdahl's Law

問題 16

2.5 / 2.5 分數

## Which of these is an example of a forbidden *inter-loop dependency*?

- ( ) a[i] = (float)( i );
- (●) a[ i ] = a[ i-1 ] + 1.;
- ( ) a[ i ] = b[ i ] + 1.;
- ( ) a[ i ] = 2.*a[ i ];

問題 17

2.5 / 2.5 分數

## The reason that our OpenMP programs have a NUMTRIES for-loop is to:

- ( ) Determine the median performance
- (●) Determine the peak performance
- ( ) Determine the standard deviation of performance
- ( ) Determine the range of performance numbers

問題 18

2.5 / 2.5 分數

**The line “#pragma omp single” is used to:**

- Force this block of code to undergo a single reduction
- Force this block of code to be divided up into individual OpenMP sections
- Force this block of code to be executed by one thread only
- Force this block of code to be executed in single-file order by each thread

問題 19

2.5 / 2.5 分數

**The cache that is smallest and fastest is named:**

- L2
- L1
- L3
- L0

問題 20

2.5 / 2.5 分數

**The cache that is closest to the Arithmetic Logic Unit (ALU) is named:**

- L0
- L3
- L2
- L1

問題 21

2.5 / 2.5 分數

**In CS 475/575, the maximum number of Bonus Days that you can use on any one projects is:**

- 2
- 5
- 4
- 3

問題 22

2.5 / 2.5 分數

## One way to prevent harm from race conditions is:

- Dynamic scheduling
- Private variables
- Shared variables
- Mutual Exclusion Locks

問題 23

2.5 / 2.5 分數

## Coarse-grained parallelism is:

- Dividing the problem into a large number of small pieces
- Dividing the problem into a small number of large pieces
- Dividing the problem into equal-size pieces
- Dividing the problem into pieces, of all which have to be a different size

問題 24

2.5 / 2.5 分數

## In multithreading, the threads all share:

- Heap, Global variables, and the same Stack
- Execution instructions, Global variables, and the same Stack
- Heap, Execution instructions, and the same Stack
- Heap, Execution instructions, and Global variables

問題 25

2.5 / 2.5 分數

## A "Mutex" is:

- A "multiple texture" for graphics processing
- A "mutual text" message
- A sound you make when you sneeze
- Another term for a "mutual exclusion lock"

問題 26

2.5 / 2.5 分數

## A Barrier is:

- A place in the code where the first thread to get there issues an interrupt
- A place in the code that all threads must reach before any of them are allowed to continue
- A place in the code that threads are not allowed to pass ever
- A place in the code where threads can spawn other threads

問題 27

2.5 / 2.5 分數

## False Sharing happens because:

- One core is writing to a cache line at the same time another core is reading or writing the same cache line
- Two cores are not sharing the same cache line, but should be
- One core is writing to a cache line at the same time another core is reading or writing a different cache line
- Two cores are reading from the same cache line

問題 28

2.5 / 2.5 分數

When adding up the elements of a 2D array in C or C++, it is faster to add the elements:

- Vetically (i.e., down the columns) first
- Horizontally (i.e., across the rows) first
- It makes no speed difference either way

問題 29

2.5 / 2.5 分數

## SPMD stands for:

- Single Program, Multiple Data
- Significant Parallelism, Multiple Data
- Single Program, Much Data
- Significant Parallelism, Much Data

問題 30

2.5 / 2.5 分數

## How many total Bonus Days are you allowed in CS 475/575?

- 3
- 2
- 5
- 4
- 6

問題 31
2.5 / 2.5 分數

**The theoretical maximum speedup that you can *ever* achieve, no matter how many cores you add, is:**

- 1/(Fp+Fs)
- Fs
- 1/(1-Fp)
- 1/Fp

問題 32
2.5 / 2.5 分數

**A Chunksize of 2:**

- Breaks your array into 2 even pieces
- Uses two threads only
- Deals two for-loop passes to each thread and then goes around to each thread again, etc.
- Breaks your array into 2 uneven pieces

問題 33
2.5 / 2.5 分數

**The difference between using OpenMP Tasks vs. using OpenMP Sections is that:**

- Nothing -- they are different words for the same thing
- Tasks are statically allocated, sections are dynamic
- Tasks are dynamically allocated, sections are static
- Sections are deprecated

問題 34
2.5 / 2.5 分數

## The advantage of using the OpenMP *reduction* clause is

- It is less likely to result in a compiler error
- No advantage, it is just cleaner code
- Actually a disadvantage -- it can produce wrong, non-deterministic answers
- It greatly speeds, and makes thread-safe, reduction operations

問題 35
2.5 / 2.5 分數

## Gustafson's Observation on Amdahl's Law says:

- When people buy more cores they often do it to process more data, which results in a larger parallel fraction
- When people buy more cores they often do it to reduce memory contention, which decreases performance
- Amdahl's Law only applies when you have a number of cores that is less than or equal to 8
- Amdahl's law was applicable when it was formulated, but doesn't apply now

問題 36
2.5 / 2.5 分數

## Using "default(none)" in an OpenMP #pragma is:

- A good idea, but not required
- Required
- A deprecated feature of an older version of OpenMP
- A way to possibly increase performance

問題 37
2.5 / 2.5 分數

## A Monte Carlo probability is computed by:

- Dividing the number of successes by the number of trials
- Dividing the number of trials by the number of successes
- Adding the number of successes to the number of trials
- Subtracting the number of successes from the number of trials

問題 38
2.5 / 2.5 分數

## Intel recently broke the CPU clock speed record by:

- Cooling the chip with liquid FlourInert
- Cooling the chip with four fans
- Cooling the chip with liquid helium
- Running the CPU outside during a colder-than-usual winter

問題 39
2.5 / 2.5 分數

## Speedup Efficiency is defined as:

- Sn/n
- Fp/n
- Fp
- n

問題 40
2.5 / 2.5 分數

## The *observation* that *clock speed* doubles every 2 years:

- Is only correct for CPUs, not GPUs
- Was never actually observed on real systems
- Has been correct starting in 1965 and is still happening
- Was the case for a while, but does not apply anymore

測驗分數： 得分：100；總分：100