

科目：作業系統 A

日期：112 年 7 月 5 日 第 1 頁 共 2 頁

請 “✓” 明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (3%) A paging system uses a 2-level page table with an 95% TLB hit ratio. Assume that the TLB search time is ignorable comparing with the memory access time. What is the effective average memory access time if the memory access time is 1 micro second?
2. (9%) A disk has 200 cylinders that are numbered from 0 to 199. Consider a disk queue holding requests to the following cylinders in the listed order: 116, 22, 3, 11, 75, 185, 100, 87. Assume that the disk head is currently at cylinder 50 and moving upward through the cylinders for serving requests. Please give the order of cylinders that the disk head will stop by using the SCAN, C-SAN, and LOOK scheduling algorithms, respectively.
3. (6%) Please describe when priority inversion will occur? How to solve it?
4. (4%) Given a UNIX i-node with ten direct blocks, one single index block, one double index block and one triple index block. Assume that the size of a block and the block address are 16Kbytes and 4 bytes, respectively. What would be the size of the largest file allowed in byte?
5. (4%) How many “hello” does the following code print if no error occurs?

```
main (int argc, char **argv) {  
    int i;  
    for (i=0; i < 4; i++) {  
        fork();  
        printf("hello\n", getpid());  
    }  
}
```
6. (3%) Please briefly describe the Direct Memory Access (DMA)?
7. (3%) What is the race condition?
8. (6%) What are the internal and external fragmentations?

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：作業系統 A

日期：112 年 7 月 5 日 第 2 頁 共 2 頁

The following are multi-choice questions.

9. (2%) The File Allocation Table of Microsoft's FAT32 file system is a variation of:
- (a) linked allocation.
 - (b) contiguous allocation.
 - (c) indexed allocation.
 - (d) combined indexing.
10. (2%) Page fault frequency monitoring is a technique for:
- (a) evaluating the effectiveness of the MMU.
 - (b) evaluating the effectiveness of the TLB.
 - (c) managing the resident set for all processes in the system.
 - (d) optimizing the interrupt response time for page faults.
11. (2%) A semaphore puts a thread to sleep:
- (a) if it tries to decrement the semaphore's value below 0.
 - (b) if it increments the semaphore's value above 0.
 - (c) until another thread issues a notify on the semaphore.
 - (d) until the semaphore's value reaches a specific number.
12. (2%) Which component of a process is not shared across threads?
- (a) Program memory.
 - (b) Heap memory.
 - (c) Global variables.
 - (d) CPU Registers.
13. (2%) A system has 32-bit logical addresses and 1 gigabyte main memory with page size of 256 Kbytes. How many entries of an inverted page table would contain?
- (a) 1K entries
 - (b) 4K entries.
 - (c) 16K entries.
 - (d) 32K entries.
14. (2%) Thrashing occurs when:
- (a) The scheduler flip-flops between two processes, leading to the starvation of others.
 - (b) The sum of the working sets of all processes exceeds available memory.
 - (c) Two or more processes compete for the same region of shared memory and wait on mutex locks.
 - (d) Multiple processes execute in the same address space.

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：作業系統 B

日期：112 年 7 月 5 日 第 1 頁 共 2 頁

請 “✓” 明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. [3pts] Please briefly describe the role of the operating system.
2. [3pts] Please point out three challenges when a system does not run an operating system on it.
3. [3pts] Please briefly describe the purpose of system calls.
4. [3pts] Please briefly describe the purpose of virtual memory.
5. [3pts] Please describe what kind of data will be stored in slabs.
6. [3pts] Please point out two situations where the system will run the context switch.
7. [3pts] Please briefly describe the purpose of virtual file system.
8. [5pts] In Linux, kernel allocates different amount of time slice to processes with different priority levels (or Nice values), as shown in the table below. Please point out two challenges for low-priority processes (i.e., processes have a higher nice value) under this allocation strategy.

Nice Value	-20	-19	-18	-17	-16	-15	-14
Time Slice (ms)	800	780	760	740	720	700	680
Nice Value	13	14	15	16	17	18	19
Time Slice (ms)	35	30	25	20	15	10	5

9. [4pts] Modern x86 systems adopt multi-level page tables. How many levels are there on the system where the page size, virtual address space size and the page table entry size are set to 2MB, 2^{64} B, 8B, respectively? (Please show the calculation process. Will get no points without the calculation process.)
10. [20pts] Please answer **True** or **False** for each statement below:
 - a. The process stack contains temporary data (such as function parameters and local variables) and the process heap is allocated dynamically.
 - b. When a user process executes a system call, the system runs context switch to switch-in the kernel process and executes the corresponding kernel code.
 - c. All processes share the same page table.
 - d. User threads execute within the same program in a shared memory address space, and thus they share open files, heap and page table.
 - e. Paging involves breaking physical memory into fixed-sized blocks called frames and breaking logical memory into blocks of the same size called pages
 - f. Systems always run context switches after handling page faults.

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：作業系統 B

日期：112 年 7 月 5 日 第 2 頁 共 2 頁

- g. Different files in the file system can share the inode number.
- h. Two processes may have the same file descriptor number, and the descriptor can point to different files.
- i. Operating system can access physical addresses directly without using the virtual memory.
- j. CPU can directly access the data in the hard disk drive without the help from operating system.

科目：計算理論 A

日期：112 年 7 月 4 日 第 1 頁 共 1 頁

請 "✓" 明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (10%) Prove or disprove that $L = \{ 0^k u 0^k \mid k \geq 1 \text{ and } u \in \Sigma^* \}$, where $\Sigma = \{0, 1\}$, is regular.
2. (10%) Prove or disprove that $L = \{ 0^k 1 u 0^k \mid k \geq 1 \text{ and } u \in \Sigma^* \}$, where $\Sigma = \{0, 1\}$, is regular.
3. (10%) Prove or disprove that if A and B are both regular languages, $A \cap B$ is a regular language.
4. (10%) Prove or disprove that if A and B are both context-free languages, $A \cap B$ is a context-free language.
5. (10%) Prove or disprove that if A and B are both non-context-free languages, $A \cap B$ is a non-context-free language.

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：計算理論 B

日期：112 年 7 月 4 日 第 1 頁 共 1 頁

請 “✓” 明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. Let $L = \{ \langle M \rangle : M \text{ is a Turing machine and } \{00, 11\} \subseteq L(M) \}$, where $\langle M \rangle$ is the standard encoding of Turing machine M .
 - (a) (10%) Show that L is not recursive by the reduction method.
 - (b) (10%) Show that L is recursively enumerable.
2. This problem is about NP-completeness.
 - (a) (5%) What is the complexity class NP?
 - (b) (5%) What is an NP-complete problem?
 - (c) (20%) Let SUB-GRAPH ISOMORPHISM (SI) problem be: given two graphs $G=(V_1, E_1)$ and $H=(V_2, E_2)$, determine whether G contains a sub-graph isomorphic to H . That is, there are subsets $V \subseteq V_1$ and $E \subseteq E_1$ with $|V|=|V_2|$ and $|E|=|E_2|$, and a one-to-one function $f: V \rightarrow V_2$ satisfying $(u, v) \in E$ if and only if $(f(u), f(v)) \in E_2$ for all u and v in V . Show that the problem SI is NP-complete.

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

國立陽明交通大學 試 題 紙

——學年度第二學期
博士班資格考

科目：計算機架構 A

日期：112 年 7 月 4 日 第 1 頁 共 2 頁

請“✓”明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (8%)

- (a) (4%) Describe the *weighted arithmetic mean* and *geometric mean* for summarizing the performance results of a benchmark suite and explain in which circumstance each of them is more suitable to be applied.
- (b) (4%) Define the following two main measures of dependability: *module reliability* and *module availability*. Write the equations with respect to *mean time to failure* (MTTF) and *mean time to repair* (MTTR) for measuring these two quantities of a non-redundant system with repair.

2. (8%) A 4-GHz processor was used to execute a benchmark program with the instruction mix and clock cycle counts shown in the following table.

Instruction type	Frequency	CPI
Integer ALU ops	20%	1
Floating-point ops	40%	20
Loads/Stores	25%	6
Branches	15%	2

- (a) (3%) Assume that the total number of instructions executed is 6×10^9 , determine the *effective CPI*, and *execution time* of this program.
- (b) (3%) Assume that an enhancement proposal is to reduce the CPI of the floating-point operations to 7.5 with 20% lengthening of the clock cycle time. Calculate the *effective CPI* and the *speedup* of the enhancement.
- (c) (2%) Calculate the *fraction of the time* for executing load and store operations of this program in the original processor. Assume that we build an optimizing compiler to discard two thirds ($2/3$) of the Load/Store operations from the original instruction mix, determine the *speedup* of the enhancement to the original design by applying **Amdahl's Law**.

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：計算機架構 A

日期：112 年 7 月 4 日 第 2 頁 共 2 頁

3. (7%)
- (a) (3%) Describe the following three **block placement** approaches for **cache** and the **block identification** method corresponding to each approach: *direct mapped*, *set associative*, and *fully associative*.
 - (b) (4%) Describe the policies adopted for *block placement*, *block identification*, *block replacement*, and *write strategy* of a **paged virtual memory** and explain the reasons.
4. (6%) For a computer system, the processor is in-order execution that runs at 2 GHz and has a CPI (cycles per instruction) of 1.4 excluding memory stalls. Suppose that in 1000 memory references there are 50 misses in the first-level (L1) cache and 25 misses in the second-level (L2) cache. Assume that the hit time of L1 cache is 1 clock cycle, the hit time of the L2 cache is 20 ns, the miss-penalty from the L2 cache to memory is 200 ns, and there are 1.25 memory references per instruction. Assume that the miss penalty need not be rounded to an integral number of clock cycles. Ignore the impact of writes.
- (a) (2%) What are the *local* and *global miss rates* for each of the L1 cache and the L2 cache?
 - (b) (4%) Calculate the *average memory access time* in ns and the **overall CPI**, including memory stalls.
5. (10%) Typically, **miss rate**, **miss penalty**, **hit time**, and **bandwidth** are the main metrics to evaluate the performance of a cache memory. Describe each of the following cache optimizations and indicate its positive impact to the performance metrics:
- (a) *way-prediction cache*
 - (b) *pipelined cache*
 - (c) *merging write buffer*
 - (d) *blocking* (a compiler optimization)
 - (e) *hardware prefetching*
6. (11%)
- (a) (3%) Describe the following three variations of **SIMD**: *vector architectures*, *multimedia SIMD instruction set extensions*, and *graphics processing units (GPUs)*.
 - (b) (8%) Describe each of the following optimizations of **vector architecture** and the problem it solves:
 - i. *vector-length register* ii. *predicate register* iii. *stride* iv. *gather-scatter*

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：計算機架構 B

日期：112 年 7 月 4 日 第 1 頁 共 2 頁

請 "✓" 明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (10%) The branch predictors are crucial for the performance of high-performance CPUs. Answer the following questions about dynamic branch predictors:
 - (a) (2%) What is an (m, n) two-level predictor?
 - (b) (1%) For a gshare branch predictor, we must record the global branch history and the local history of a branch instruction. What kind of circuit component do we usually use to record the global branch history?
 - (c) (2%) How do we compose the index to the branch prediction buffer for a gshare branch predictor?
 - (d) (5%) Please describe the key design principles of the TAGE predictor?
2. (10%) Dynamic scheduling is an important technique for modern high-performance out-of-order superscalars.
 - (a) (5%) Please give three reasons why dynamic scheduling is important for high performance CPUs.
 - (b) (5%) Among the three types of data dependences (hazards), which ones mainly occurs due to dynamic scheduling of the processor cores (assume that the ISA is RISC-based)? How do you resolve such types of dependences without stalling?
3. (10%) For out-of-order execution, we can use the simple scoreboarding technique of a more sophisticated scheduling technique such as the Tomasulo's algorithm.
 - (a) (2%) What is the main differences between scoreboarding and Tomasulo's algorithm.
 - (b) (3%) What are the reservation stations in Tomasulo's algorithm? You must explain their usage in detail.
 - (c) (5%) Please draw a block diagram that shows the basic structure of the Tomasulo's algorithm for executing floating-point instructions with multiple adders and multipliers, but without speculations.
4. (5%) Exception handling is complicated for out-of-order processors.
 - (a) (2%) What is an imprecise exception?
 - (b) (3%) What are the two main causes of an imprecise exception?

科目：計算機架構 B

日期：112 年 7 月 4 日 第 2 頁 共 2 頁

5. (3%) Please use a table to list the following characteristics, including the issue structure, the scheduling technique, and the instruction execution order, for the following three multiple-issue architecture: static superscalar, speculative superscalar, and VLIW.
6. (8%) For multicore or multi-processor systems, coherent data caches are crucial to its performance.
 - (a) (4%) What is the MESI protocols? You must explain each state in the protocol in your answer. What is the main advantage of this protocol compared to the simple three-state coherence protocol?
 - (b) (4%) Why do modern high-performance processors usually includes an L3 cache, instead of simply increasing the size of the L2 cache, in order to increase the performance of thread-level parallelism for data accesses? For a memory-demanding application, what are the main causes of L3 cache misses of a very large L3 caches?
7. (4%) Modern high-performance processors typically support fence instructions to ensure thread synchronization. Please explain what a fence instruction is. Why do we need such instructions instead of simply letting the processor to enforce a fence at each memory operation?

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：演算法 A

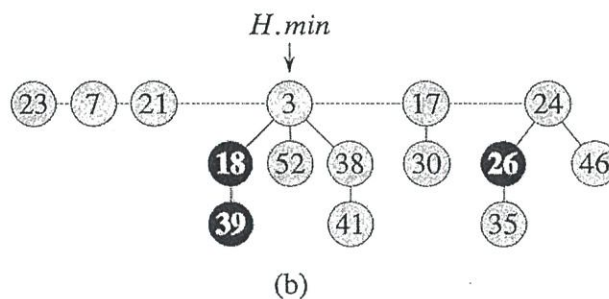
日期：112 年 7 月 5 日 第 1 頁 共 1 頁

請 "✓" 明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (a) Given $\sum_{k=0}^n x^k = \frac{x^{n+1}-1}{x-1}$, show that $\sum_{k=0}^n x^k \leq \frac{1}{1-x}$ if $|x| < 1$. 5%
 (b) Name an application that we need the above inequality in the time complexity analysis of the algorithm. 5%
2. Fibonacci Heap:



- (a) Decrease key of node that has value 41 to a new value 4, then 4%
 (b) Delete minimum from the resulted Fibonacci Heap. 7%
3. About insertion sort an array A of n integers:
 - (a) Prove by induction that insertion sort correctly sorts n integers. 6%
 (b) Let a_i be an item in A , and k_i be the rank of a_i (means that a_i is the k_i th largest integer among all integers in the array). We don't know what k_i is, but we do know that a_i stored in $A[I_i]$ and $|k_i - I_i| \leq 10$. What can you conclude about the time required to insertion sort the n integers? 6%
4. Design a heap (storing n integers) that supports the operations, insert arbitrary integer and delete the k th largest integers, all in $O(\log n)$ time. Describe the data structure, and use pseudo code to describe the operations `Insert (int)`, and `Del_kth_largest()`. 12%
5. Dynamic table, insertion only, expand the table size when we try to insert a new item into a full table: Show that each insertion can be done in constant amortized cost. 5%

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：

科目：演算法 B

日期：112 年 7 月 5 日 第 1 頁 共 1 頁

請 “✓” 明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (10%) Can all problems in NP be solved in time $O(n^{2023})$ if we could solve an NP-complete problem in time $O(n^{2023})$? Briefly justify your answer.
2. (10%) Let $G = (V, E)$ be a connected undirected weight graph with strictly positive weights $w: E \rightarrow \mathbb{Z}^+ = \{1, 2, 3, \dots\}$ where $|E| = |V| = n$. Describe an $O(n)$ -time algorithm to determine a path from vertex u to vertex v of minimum weight.
3. (15%) The NP-complete SUBSET-SUM problem asks, given a set of non-negative integers S and a target k , whether or not a subset from a list of integers can sum to a target value. For example, consider the list of $S = \{1, 2, 3, 4\}$. If the target $k = 6$, then there are two subsets that achieve this sum $\{1, 2, 3\}$ and $\{2, 4\}$. If $k = 12$, then there are no solutions. The AVERAGE-SUM problem asks, given just a set of non-negative integers S , whether or not a subset from a list of integers can sum to the average of S (i.e. $\frac{1}{|S|} \sum_{i \in S} i$ where $|S|$ is the number of elements in S). It is similar to the SUBSET-SUM problem, except now the target value is always the average value in S . Prove that AVERAGE-SUM is NP-complete, or give a polynomial-time algorithm for it.
4. (15%) Bill is having dinner at a bar, where he will order many small plates. There are n plates of food on the menu, where information for plate i is given by a triple of non-negative integers (v_i, c_i, s_i) : the plate's volume v_i calories c_i and sweetness $s_i \in \{0, 1\}$ (the plate is sweet if $s_i = 1$ and not sweet if $s_i = 0$). Now, he wants to eat as much as possible, but no more than k calories this time. He also wants to order exactly $s < n$ sweet plates, without purchasing the same dish twice. Please solve this problem using dynamic programming.
 - (a) Identify the set of subproblems. [5%]
 - (b) Identify the relationship between subproblems. [5%]
 - (c) Briefly discuss the time complexity of your DP algorithm. [5%]