

科目：作業系統 A

日期：114 年 7 月 2 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. (4%) What is "local replacement" and "global replacement"?

2. (3%) How many times does this code print "hello"?

```
main () {  
    int i;  
    for ( i=0; i < 3; i ++ ) {  
        fork();  
        execl ("/ bin/echo", "echo", "hello", 0);  
    }  
}
```

3. (4%) What are the parameters of Multilevel Feedback Queue Scheduling? (List at least four)

4. (8%) A paging system uses a 2-level page table with an 85% 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 comparing with the memory access time?

5. (8%) An inode-based file system uses 8 Kbyte blocks with 4-byte block address. What is the largest file size that the file system can handle if an inode has 12 direct blocks, 1 indirect block, and 1 double indirect block?

6. (10%) Given a disk with 200 tracks, numbered from 0 to 199, Assume the disk head is initially located at track 42 and was moving in the direction of increasing track number. Let the requested tracks, in the order received by the disk scheduler, be 116, 22, 3, 11, 75, 185, 100, 87, and 186. What is the order that the requests are serviced by using the First Come First Serve(FCFS), Shortest Seek Time First (SSTF), SCAN, C-SCAN, and LOOK scheduling algorithms, respectively?

7. (3%) What is the race condition?

8. (2%) A memory management unit (MMU) is responsible for:

- (a) Translating a process' memory logical addresses to physical addresses.
- (b) Allocating kernel memory.
- (c) Allocating system memory to processes.
- (d) All of the above.

科目：作業系統 A

日期：114 年 7 月 2 日 第 2 頁 共 2 頁

9. (2%) A major problem with the base & limit approach to memory translation is:

- (a) It requires the process to occupy contiguous memory locations.
- (b) The translation process is time-consuming.
- (c) The translation must be done for each memory reference.
- (d) A process can easily access memory that belongs to another process.

10. (2%) Which of the following does NOT cause a trap?

- (a) A user program divides a number by zero.
- (b) The operating system kernel executes a privileged instruction.
- (c) A programmable interval timer reaches its specified time.
- (d) A user program executes an interrupt instruction.

11. (2%) Which type of user thread and kernel thread mapping cannot take advantage of a multiprocessor?

- (a) 1:1
- (b) N:1
- (c) N:M
- (d) 1:M

12. (2%) Which of the following CPU scheduling algorithms is non-preemptive scheduler?

- (a) Shortest Job First (SJF)
- (b) Shortest Remaining Time First (SRTF)
- (c) Round Robin (RR)
- (d) Rate Monotonic

科目：作業系統 B

日期：114 年 7 月 2 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. [5pts] Please elaborate on the procedure for processes context switching in detail.
2. [5pts] Why does process creation adopt Copy-on-Write (CoW)? How does CoW work?
3. [5pts] Interrupt handlers usually comprise two parts, that is top halves and bottom halves. Please briefly describe the purpose of each part.
4. [5pts] Modern x86 systems usually adopt multi-level page tables. What is the minimum levels are there for the system where the page size, virtual address space size and page table entry size are set to 4KB, 2^{64} B, and 64B, respectively.
(Please show the calculation process. No points without the calculation process.)
5. [30pts] Please answer **True** or **False** for each statement below:
 - a. When a user process executes a system call, the system runs the process-level context switch to switch-in the kernel process and execute the corresponding kernel code.
 - b. A thread shares with other threads belonging to the same process its code section, file descriptors and signals.
 - c. When an OS adopts Copy-on-Write (CoW) to fork a process, memory pages will be shared between the parent and child processes until one of them writes to the page.
 - d. Upon detecting a signal asserted by a peripheral device on the interrupt- request line, the CPU executes the corresponding interrupt-handler service routine located at a predetermined memory address.
 - e. If a required data is not in DRAM, the user process will raise a page fault, and the user process will run the page fault handler to transfer the data from storage to memory, possibly using DMA for efficiency.
 - f. After enabling virtual memory, CPUs in the user mode can decide to access physical addresses or virtual addresses based on the compiler.
 - g. Physical memory refers to the actual memory location of data, while virtual memory represents the data address in the CPU cache.

科目：作業系統 B

日期：114 年 7 月 2 日 第 2 頁 共 2 頁

- h. On the operating system adopting paging (e.g., Linux), multiple processes can share one page table to save memory consumption.
- i. OS kernel can allocate a virtual memory area (VMA) to user processes, but the page frames will be allocated when the VMA is accessed.
- j. When an x86 CPU encounters a logical address that results in a TLB miss, the CPU doesn't require the execution of Linux to walk the page table for translating to a physical address.
- k. In Linux file systems, directory blocks store file names and inode references, while file content is stored separately from directory blocks.
- l. To get better performance, Linux systems cache file data, heap and stack data inside the page cache in the DRAM.
- m. File-name extensions (e.g., .txt, .jpeg) on UNIX systems are necessary and critical, as they guide the operating system to run the right application (e.g., text editor or gallery) for opening files.
- n. To access an open file, the process will receive a file descriptor and check both dentry and inode objects to retrieve the file content.
- o. Using the standard C library, a user program can easily read from or write to files without direct assistance from the operating system.

科目：計算理論 A

日期：114 年 7 月 1 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. (15%) Show that the languages $A = \{ \langle F \rangle \mid F \text{ is a DFA over } \Sigma = \{0,1\} \text{ that accepts some string of form } ww^R \}$ is Turing-decidable.

2. (15%) Show that the language

$D = \{ \langle M_1, M_2, M_3 \rangle \mid M_1, M_2, M_3 \text{ are TM's and } L(M_i) \cap L(M_j) \neq \emptyset \text{ for some } 1 \leq i \neq j \leq 3 \}$ is TM-recognizable.

3. (20%) Show that the language

$UselessState_{TM} = \{ \langle M, q \rangle \mid M \text{ is a TM and } q \in q \text{ is a useless state} \}$ is TM-undecidable, where a state q of M is useless if it cannot be reached from the start state of M by reading any input. Hint: reduce \overline{HALT} to it.

國立陽明交通大學 試 題 紙

一一三學年度第二學期
博士班資格考

科目：計算理論 B

日期：114 年 7 月 1 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 if L_1 and L_2 are both regular languages, the intersection of L_1 and L_2 is also regular.
2. (10%) Prove or disprove that if L_1 and L_2 are both context-free languages, the intersection of L_1 and L_2 is also context-free.
3. (10%) Prove or disprove that if L_1 is a regular language and L_2 is a context-free language, the intersection of L_1 and L_2 is context-free.
4. (10%) Prove or disprove that for $\Sigma = \{a, b, c\}$, $\{a^n b^n \mid n \in \mathbb{N}\}$ is a context-free language.
5. (10%) Prove or disprove that for $\Sigma = \{a, b, c\}$, $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ is a context-free language.

科目：計算機架構 A

日期：114 年 7 月 1 日 第 1 頁 共 8 頁

請 "✓" 明 ✓不可看書 可看書

* 請將答案直接寫在試題紙上

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. Circle the correct answers. [8 points]

- i. Which of the following are likely effects of deeper pipelining on performance and hardware complexity?
 - Increased branch misprediction penalty.
 - Higher throughput under ideal conditions
 - Increased forwarding/bypassing complexity.
 - Reduced dynamic instruction count.
- ii. In a directory-based cache coherence protocol, which of the following actions would the home node perform in response to a read miss by a processor, when another processor holds the block in Modified state?
 - Send the block from memory to the requester
 - Forward the request to the owner and update sharers list
 - Invalidate all other copies of the block
 - Downgrade the owner to Shared state and send a copy to the requester
- iii. What would happen in an out-of-order processor if the reorder buffer is removed and instructions are allowed to commit as soon as they complete execution?
 - Data hazards would be eliminated
 - Exceptions would be harder to handle correctly
 - Control hazards would disappear
 - More instructions could execute in parallel
- iv. Which of the following design changes would likely improve SMT scalability?
 - Increasing the size of the issue queue
 - Reducing L1 cache latency
 - Adding more rename registers
 - Lowering DRAM access latency

科目：計算機架構 A

日期：114 年 7 月 1 日 第 2 頁 共 8 頁

2. Dynamic Branch Prediction [10 points]

Consider the following RISC-V code. The register x0 is always 0.

Code sequence

```
addi x1, x0, 1
addi x6, x0, 5
addi x10, x0, 0
Loop_i:
    addi x2, x0, 1
    mv    x3, x1
Loop_j:
    add  x10, x10, x2
    addi x2, x2, 1
    bne  x2, x3, Loop_j    # -- Branch 1
    addi x1, x1, 1
    bne  x1, x6, Loop_i    # -- Branch 2
```

Each table below refers to a single branch. For example, Branch 1 will be executed 10 times—each execution should be recorded in the corresponding table. Similarly, Branch 2 is executed 5 times.

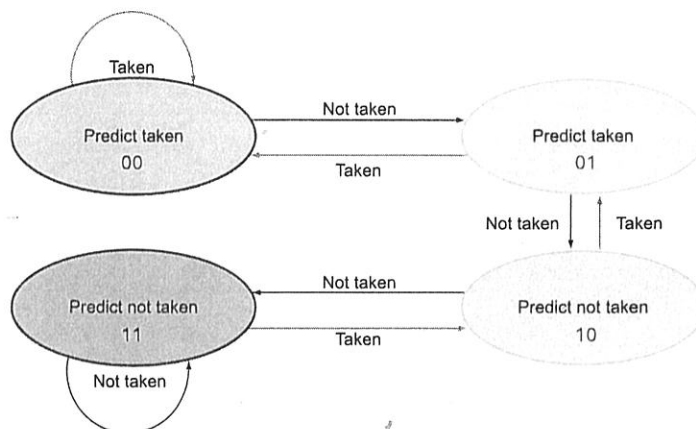
(a) Assume the processor uses **1-bit branch predictors**, both initialized to **N (Not Taken)**. Use the tables below to record the prediction and actual outcome of each branch for every dynamic execution. Some entries are provided as examples.

Step	Branch 1 Prediction	Actual Action	Branch 2 Prediction	Actual Action
1	N	N	N	T
2				
3				
4				
5				
6				
7				
8				
9				
10				

科目：計算機架構 A

日期：114 年 7 月 1 日 第 3 頁 共 8 頁

- (b) Now assume the processor uses **2-bit saturating counters** for branch prediction, each initialized to **11 (Weakly Not Taken)**. Use the state transition diagram below to update the counter values after each branch and record the corresponding predictions and outcomes. (State diagram is shown below.)



Step	Counter Value	Branch 1 Prediction	Actual Branch 1 Action	Counter Value	Branch 2 Prediction	Actual Branch 2 Action
1	11	N	N	11	N	T
2						
3						
4						
5						
6						
7						
8						
9						
10						

- (c) Calculate and compare the misprediction rates for (a) and (b). Based on your results, which predictor would you prefer to use for this loop structure, and why?

科目：計算機架構 A

日期：114 年 7 月 1 日 第 4 頁 共 8 頁

3. ISA Principles [10 points]

Consider the design of a new RISC-style ISA intended for high-performance embedded systems with aggressive energy constraints. You are tasked with evaluating several competing ISA-level design options. Answer the following:

(a) You are given two encoding formats for immediate values:

- Format A: 32-bit instruction with 12-bit immediate
- Format B: 16-bit compressed instruction with 6-bit immediate

Compare the trade-offs of using Format B in terms of:

- Instruction density
- Decoding complexity
- Performance (CPI and cache behavior)
- Suitability for control-intensive vs data-intensive workloads

(b) Consider a proposed ISA extension that adds predicated execution (e.g., executing instructions conditionally without branches). Discuss how this affects:

- Pipeline control logic
- Instruction fetch bandwidth
- Speculative execution and branch prediction

科目：計算機架構 A

日期：114 年 7 月 1 日 第 5 頁 共 8 頁

4. Static Scheduling [12 points]

A VLIW processor can issue up to 2 memory references, 2 floating-point operations, and 1 integer operation or branch per cycle. You are optimizing the following scalar loop for this VLIW target:

Code sequence

```
for (i=999; i>=0; i=i-1)
    x[i] = x[i] + s;
```

Assume that x is a double-precision floating-point array and s is a scalar floating-point value. The compiler is allowed to freely reorder and unroll instructions to optimize performance. The target VLIW processor does not support dynamic scheduling and relies entirely on the compiler to expose instruction-level parallelism (ILP).

(a) Write a RISC-V style code for the above loop.

(b) Explain why unrolling and static scheduling are critical in VLIW machines, particularly for loops like the one above. What types of pipeline stalls can be avoided through aggressive unrolling and scheduling?

科目：計算機架構 A

日期：114 年 7 月 1 日 第 6 頁 共 8 頁

(c) Unroll the loop the minimum number of times to eliminate any stall. Each memory operation has a 2-cycle latency, floating-point operations have a 3-cycle latency, and integer operations have a 1-cycle latency. You have access to 32 floating-point registers (f0–f31). Fill in the following VLIW scheduling table, showing one instruction bundle per row

Memory reference 1	Memory reference 2	FP operation 1	FP operation 2	Integer operation/branch
fld f0, 0(x1)				

(d) Based on your completed schedule in part (b), answer the following:

- The issue efficiency (utilized slots / available slots)
- The average cycles per iteration of the original scalar loop
- The average cycles per result (floating-point add)

科目：計算機架構 A

日期：114 年 7 月 1 日 第 7 頁 共 8 頁

5. Coherence [10 points]

You are given a RISC-V system with four processors, each with a 1KB direct-mapped data cache and a 64-byte block size. The system uses a snoopy, bus-based MESI cache coherence protocol. All cache lines start in the Invalid state.

Each processor executes memory operations interleaved over multiple clock cycles. Track the state of each relevant cache line and its presence in the caches over time. Be sure to duplicate the information from time step to following time step if the data stays valid in the cache and show the state that the data is in.

Time	Processor 1	Processor 2	Processor 3	Processor 4
1	lw x1, 64(x0)			
2				lw x2, 72(x0)
3				sw x3, 128(x0)
4			sw x6, 80(x0)	
5		lw x2, 72(x0)		
6		sw x3, 1096(x0)		
7		lw x4, 1088(x0)		
8	sw x3, 64(x0)			
9	sw x4, 128(x0)			
10			lw x1, 32(x0)	

Time	Processor 1		Processor 2		Processor 3		Processor 4	
	Line address	Line state	Line address	Line state	Line address	Line state	Line address	Line state
1								
2								
3								
4								

國立陽明交通大學 試 題 紙

一一三學年度第二學期
博士班資格考

科目：計算機架構 A

日期：114 年 7 月 1 日 第 8 頁 共 8 頁

Time	Processor 1		Processor 2		Processor 3		Processor 4	
	Line address	Line state	Line address	Line state	Line address	Line state	Line address	Line state
5								
6								
7								
8								
9								
10								

科目：計算機架構 B

日期：114 年 7 月 1 日 第 1 頁 共 3 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 program is composed of ALU, load, store, and branch instructions. The percentage of each instruction type and cycles spent in each instruction are shown below.

Instructions	Percentages of instructions	Cycles per instructions
ALU	40%	2
Load	25%	10
Store	15%	5
Branch	20%	4

- (a) What is the average CPI when a processor executes such a program? (2pt)
- (b) A computer architect designs a new processor that makes 80% of ALU instructions take only 1 cycle to execute. The other 20% of ALU instructions will still take 2 cycle to execute. In addition, the computer architect makes 90% of load instructions to complete in 2 cycles while the remaining 10% of load instructions take 10 cycles to execute per load. Finally, the computer architect also improves the store instructions in such a way that each instruction takes 2 cycles to execute. What is the average CPI in this new processor? (3 pt)
2. The clock frequency of a processor is 2 GHz and its Cycle Per Instruction (CPI) is 1.4 without including the stall cycles of cache misses. In addition, this processor has an I-cache and a D-cache. The cache hit time is 1 clock cycle. The I-cache has a 4% cache miss rate and the D-cache has a 6% miss rate on load and store instructions. The cache miss penalty is 50ns, which is the time to access and transfer a cache block (line) between main memory and the processor. Finally, load and store instructions count 30% of all instructions in a program.
- (a) What is the cache miss penalty (in cycles) in this 2GHz processor? (3 pt)
- (b) What is the number of stall cycles per instruction? (3 pt)
- (c) What is the overall CPI after considering the stall of the cache miss? (3 pt)
- (d) A computer architect improves the design of the data cache and shortens the time of load instructions by a factor of 3x (3 times faster) and store instructions by a factor of 4x. What is the overall speedup of the program after the improvements of the data cache when a program includes 30% load and 16% store instructions? (3 pt)
3. A virtual memory system that can address a total of 2^{40} bytes with 2GB physical memory. The virtual and physical pages are each 8KB in size.

科目：計算機架構 B

日期：114 年 7 月 1 日 第 2 頁 共 3 頁

- (a) How many virtual and physical pages in the system? (4 pt)
- (b) We decide to accelerate the virtual memory by using a translation lookaside buffer (TLB) with 256 entries. How big (in bits) is the TLB? (Hint: a TLB entry contains Physical Page Number (PPN), Virtual Page Number (VPN), and a valid bit) (3 pt)
- (c) Assume that TLB is organized as 2-way set associative cache, please draw out the organization of a TLB organized as the two-way set-associative cache composed of gates for cache hit and data fetch, and each set should include valid, tag, and data block field. (4 pt)

4. A program is composed of following assembly codes and is executed in a processor having a data cache with 8 cache blocks (lines). The memory address in this processor is 32 bits. What is the data cache miss rate in the variation of the data cache architectures when executing the following loop function?

	addi	s0	0	10
loop	beq	s0	0	done
	lw	s1	0x6(0)	
	lw	s2	0x26(0)	
	add	s3	s1	s2
	addi	s0	s0	-1
	j	loop		
done:				

- (a) What is the cache miss rate (in percentage) when using the direct-mapped cache? (4 pt)
- (b) What is the cache miss rate (in percentage) when using 2-way set associative cache? (4 pt)
- (c) What is the cache miss rate (in percentage) when using fully-associative cache? (4 pt)
5. The vector processor exploits single instruction multiple data (SIMD) manner to accelerate the program.
- (a) A pipelined vector processor consists of 32-element vectors, 2 lanes, and 8 cycle to complete a multiply instruction. The scalar processor takes 320 cycles ($=32 * 10$) to complete the multiply calculation of 32 elements. What is the timing (in cycles) when using this pipelined vector processor to execute 32 elements multiply? (3 pt)
- (b) A pipelined vector processor is composed of 4 element vectors and 1 lane. This vector processor takes 4-cycle for ldf.v/stf.v, 8-cycle for mulf.sv (sv is scalar-vector), and 5-cycle for addf.vv instructions. In addition, this vector processor executes the following SAXPY program that contains vector registers (v1 to v4). Please fill in the following table (grey fields) until the cycle 10 with the d* that is the dependence stall, the p* is the pipeline stall, E* is the execution of mulf.sv, and E+ is the execution of addf.vv. (7 pt)

科目：計算機架構 B

日期：114 年 7 月 1 日 第 3 頁 共 3 頁

ldf.v	v1	X(r1)	
mul.f.sv	v2	f0	v1
ldf.v	v3	Y(r1)	
add.f.vv	v4	v3	v2
stf.v	v4	Z(r1)	
addi	r1	16	r1
blti	r1	r2	0

(a) SAXPY program assembly

Cycle	1	2	3	4	5	6	7	8	9	10
ldf.v v1, X(r1)	F	D	X	M	M	M	M	W		
mul.f.sv v2, f0, v1		F	D							
ldf.v v3, Y(r1)			F							
add.f.vv v4, v3, v2										
stf.v v4, Z(r1)										

(b) Timing table of the SAXPY program on the pipelined vector processor

科目：演算法 A

日期：114 年 7 月 2 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. 8% Given n keys, you would like to build a binary search tree for the n keys. Can you build the binary search tree in less than $n \log n$ time? If your answer is yes, give me the algorithm; if the answer is no, give me your reason(s).
2. 8% Given n keys, you would like to build a max-heap for the n keys. Can you build the max-heap in less than $n \log n$ time? If your answer is yes, give me the algorithm; if the answer is no, give me your reason(s).
3. 8% The *level* of a node in a binary tree is defined as the following:
 - The root of the binary tree is *level* 0.
 - The child of a *level* i node has *level* $i + 1$.

Prove by induction that there are at most 2^i nodes in *level* i .

4. The Hamiltonian Cycle problem (HC): Given a graph $G = (V, E)$, whether we can find a cycle that visits each vertex exactly once. The Hamiltonian cycle problem is NP-complete.
The Traveling Salesman Problem (TSP): There are n cities, and each pair of cities is connected by a distance d road. You want to leave your home, visit all the cities, then go home, and you want to minimize the distance you travel.
 - (a) 8% If you want to show that TSP is NP-Hard, you should transform HC to TSP, or transform TSP to HC? Why?
 - (b) 5% How to make the transformation?
 - (c) 5% Do you think the weight of the Minimum Spanning Tree connecting all cities is less than the solution to the TSP? Why?
5. 8% Given a set of n real numbers $r_i, 1 \leq i \leq n$, we would like to find the minimum difference (minimum gap) between two reals. This problem can be solved by sorting the n real numbers in $O(n \log n)$ time, then spending linear time to find the minimum gap. Design a **divide and conquer** algorithm to find the minimum gap. What is the time required for your algorithm?

國立陽明交通大學 試 題 紙

科目：演算法 B

日期：114 年 7 月 2 日 第 1 頁 共 3 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. Movie selection problem

In a movie festival, n movies will be shown. You are given the starting time s_i and ending time t_i of the i -th movie.

***** DO THIS *****

NOTE: The time complexity of your code (for all of the following sub-problems) must be bounded by $O(n^2)$. Acceptable complexities include $O(1)$, $O(n)$, and $O(n^2)$, while complexities such as $O(n^2 \log n)$ and $O(n^3)$ are not acceptable.

(a) [5%] Determine the maximum number of movies you can watch in their entirety. Please write the pseudo-code for your solution.

HINT: Use a greedy algorithm.

(b) [5%] Now, each movie is associated with a preference score p_i ($p_i > 0$). What is the maximum total preference score you can achieve by watching a subset of movies without overlapping their playtimes? Please write the pseudo-code for your solution.

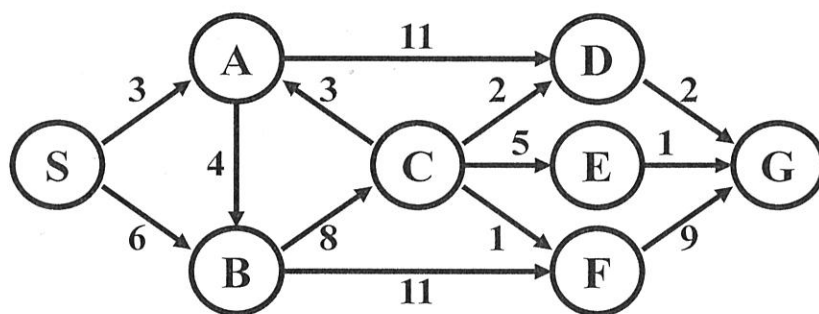
HINT: Use dynamic programming.

(c) [5%] Suppose you have a magical ability that allows you to appear in multiple places simultaneously. Therefore, you are now able to watch movies even if their playtimes overlap. However, you can only overlap movies if their indices satisfy $|i - j| = 1$ (i.e., they are adjacent in the list). Under these conditions, what is the maximum total preference score you can achieve? You do not have to write the pseudo-code for this sub-problem.

科目：演算法 B

日期：114 年 7 月 2 日 第 2 頁 共 3 頁

2. Given the following graph where vertex S is the source and the weight of each edge is specified alongside the edge, please apply the **Bellman-Ford** algorithm to find shortest paths from S to each of other vertices. During each pass/iteration of edge-by-edge relaxation, the order of edge to be processed is: $AB, AD, BC, BF, CA, CD, CE, CF, DG, EG, FG, SA, SB$ (i.e., lexicographic order, assuming that an edge from vertex X to vertex Y is named XY).



There are 8 vertices in the graph; thus, in theory, the Bellman-Ford algorithm requires up to 7 passes of edge-by-edge relaxation to reach the steady state where correct shortest paths can be found. However, if you are lucky, it sometimes requires much fewer passes.

*** DO THIS ***

- (a) [10%] Based on the aforementioned order of edge to be processed, please duplicate the graph for each pass of edge-by-edge relaxation and for each vertex X in the graph, derive $d[X]$ where $d[X]$ denotes the shortest-path weight from S to X . Duplicate as many as you want; you can stop when you reach the steady state (and the final correct value of $d[X]$ for any vertex X can be found).
- (b) [5%] In this particular graph and based on whatever bad/poor/stupid order of edge to be processed, is it possible that the Bellman-Ford algorithm requires the worst-case 7 passes to reach the steady state? Why or why not?
- (c) [5%] If the weight of edge BC is updated from 8 to -8, can you still find shortest paths from S to each of other vertices? Why or why not?

科目：演算法 B

日期：114 年 7 月 2 日 第 3 頁 共 3 頁

3. Knapsack problem

Given a set of items with specific weights and values, the goal is to get as much value into the knapsack as possible under the weight constraint of the knapsack. The problem has two different versions:

Version 1: we can get a fraction of an item; that is, items can be broken.

Version 2: we have to take an item as a whole or leave the whole item.

Consider the following example:

Item 1: weight = 10, value = 60

Item 2: weight = 20, value = 100

Item 3: weight = 30, value = 120

And the knapsack has a weight constraint of 50.

For Version 1, we will take the whole Item 1, the whole Item 2, and two-third of Item 3, to achieve the maximum value of 240 with a total weight of 50 (constraint satisfied).

For Version 2, we will take the whole Item 2 and the whole Item 3, to achieve the maximum value of 220 with a total weight of 50 (constraint satisfied).

*** DO THIS ***

- (a) [6%] Choose one between Version 1 and Version 2, whichever you think is easier to solve. Please describe your methodology for solving the problem (version) of your choice. Note that you don't have much time so make long story short.
- (b) [9%] For the other problem (version) not chosen, if your methodology proposed in (a) is applied, what would the issue be? How could the issue be resolved? Briefly describe how to resolve the issue by writing pseudo-code (with some sort of recursive formulation or similar). Here, we don't care about the time complexity of your pseudo-code implementation, but we care about our time required for grading your code. So again, make long story short.