

# 國立成功大學

## 115學年度碩士班招生考試試題

編 號：89

系 所：工程科學系

科 目：計算機概論

日 期：0204

節 次：第 2 節

注 意：1. 不可使用計算機  
2. 請於答案卷(卡)作答，於  
試題上作答，不予計分。

1. In 8-bit two's complement, compute  $0x9C + 0x36$ . Show your calculation and state the 8-bit hexadecimal result and whether overflow occurs. (5%)
2. Convert the decimal number 13.375 to binary (exact representation). Calculation or derivation is required. (5%)
3. IEEE 754 single precision bits are: sign = 0, exponent =  $10000010_2$ , fraction =  $0100000000000000000000_2$ . What is the decimal value? Calculation or derivation is required. (5%)
4. A CPU executes a program with Instruction Count =  $1.2 \times 10^9$ , average CPI = 1.5, and clock rate = 3.0 GHz. Estimate the CPU time. Calculation or derivation is required. (5%)
5. By Amdahl's Law, 40% of the execution time can be accelerated by a factor of 3. What is the overall speedup (approx.)? Calculation or derivation is required. (5%)
6. A direct-mapped cache has cache size 16 KB, block size 64 B, and 32-bit byte addressing. How many index bits and tag bits are used? Calculation or derivation is required. (5%)
7. A 32-bit virtual address space uses 4 KB pages and a single-level page table. Each page table entry (PTE) is 4 bytes. What is the page table size per process? Calculation or derivation is required. (5%)
8. A link has bandwidth 10 Mbps, distance 2000 km, and propagation speed  $2 \times 10^8$  m/s. Packet size is 1500 bytes. Ignoring queuing and processing delays, what is the one-way delay (transmission + propagation)? Calculation or derivation is required. (5%)
9. RAID-5 parity uses XOR. If  $D0 = 0x3A$  and  $D1 = 0xC7$ , what is parity  $P = D0 \text{ XOR } D1$ ? Calculation or derivation is required. (5%)
10. An algorithm performs  $T(n) = 3n^2 + 10n \log_2 n + 500$  operations. What is the tightest asymptotic bound  $\Theta(\cdot)$ ? Calculation or derivation is required. (5%)
11. Karnaugh Map and Logic Minimization (10%)

Given the 4-variable Boolean function  $F(A, B, C, D) = \sum m(0, 2, 3, 5, 7, 8, 10, 11, 13, 15)$ . Fill the K-map below with 1s and 0s (minterms not listed are 0), perform grouping, and derive the minimal SOP expression. Then estimate the minimum number of 2-input NAND gates needed to implement your minimized form (assume inverters may be implemented as NAND with tied inputs).

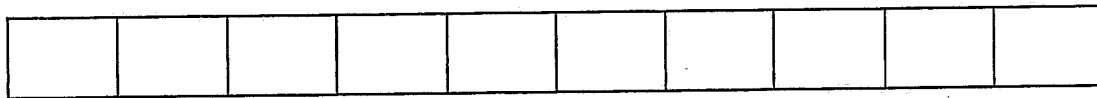
|                       |    |                 |    |    |    |
|-----------------------|----|-----------------|----|----|----|
|                       |    | AB (Gray order) |    |    |    |
|                       |    | 00              | 01 | 11 | 10 |
| CD<br>(Gray<br>order) | 00 |                 |    |    |    |
|                       | 01 |                 |    |    |    |
|                       | 11 |                 |    |    |    |
|                       | 10 |                 |    |    |    |

12. CPU Scheduling with a Gantt Chart (10%)

Four processes have the following arrival times and CPU bursts (in ms): P1: arrival 0, burst 7; P2: arrival 2, burst 4; P3: arrival 4, burst 5; P4: arrival 6, burst 3. Use Round Robin scheduling with time quantum = 3 ms. Ignore context-switch overhead. Draw the Gantt chart and compute the average waiting time and average turnaround time.

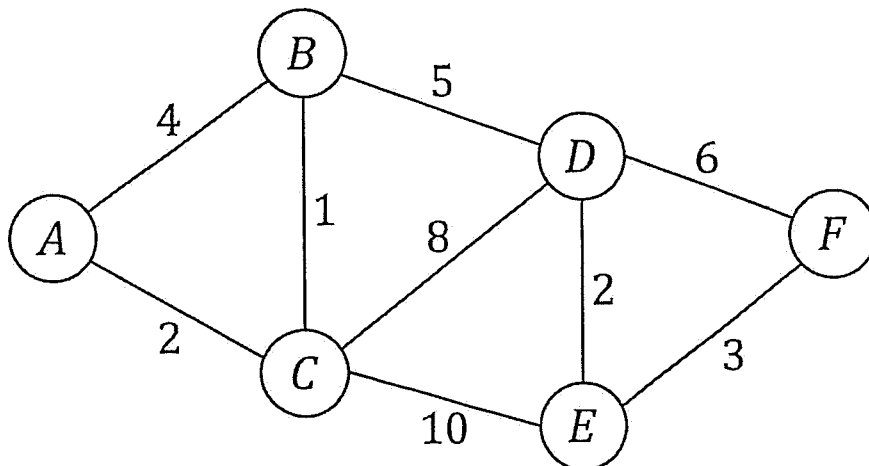
Gantt Chart Template (label time on the axis)

Write  $P_i$  in each slot; add time marks below.



13. Shortest Path on a Weighted Graph (10%)

Using Dijkstra's algorithm, find the shortest path from A to F and the final shortest distances  $dist(\cdot)$  from A to every node. Show key relaxation steps or a distance table.



14. Binary Search Tree (BST) Construction and Traversal (10%)

Insert the following keys into an initially empty BST in the given order:

50, 30, 70, 20, 40, 60, 80, 65, 35.

Draw the final BST and write the preorder, inorder, and postorder traversal sequences.

15. Cache Address Breakdown and Hit/Miss Analysis (10%)

Consider a direct-mapped cache with cache size = 1 KB, block size = 16 B, using 16-bit byte addressing.

(1) Determine the number of offset bits, index bits, and tag bits. (2%)

(2) For the following sequence of accesses (hex), determine hit or miss for each and compute the hit rate. (6%)

0x0000, 0x0004, 0x0010, 0x0100, 0x0008, 0x0110, 0x0014, 0x0104.

(3) After the final access, list each cache line index that was used and the tag stored in that line. (2%)

| Access # | Address | Tag | Index | Offset | Hit/Miss |
|----------|---------|-----|-------|--------|----------|
| 1        | 0x0000  | --- | ---   | ---    | -----    |
| 2        | 0x0004  | --- | ---   | ---    | -----    |
| 3        | 0x0010  | --- | ---   | ---    | -----    |
| 4        | 0x0100  | --- | ---   | ---    | -----    |
| 5        | 0x0008  | --- | ---   | ---    | -----    |
| 6        | 0x0110  | --- | ---   | ---    | -----    |
| 7        | 0x0014  | --- | ---   | ---    | -----    |
| 8        | 0x0104  | --- | ---   | ---    | -----    |