國立成功大學
115學年度碩士班招生考試試題

編　號：140

系　所：電機資訊學院-資訊聯招

科　目：計算機數學

日　期：0203

節　次：第 3 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。

一、Discrete Mathematics (50%) **You should show how to get the answers in detail or obtain no credit.**

1.(10%) Consider all simple (loop-free), undirected graphs on a set of **5 distinct labeled vertices** $V = \{1,2,3,4,5\}$. Every edge connects two distinct vertices, and multiple edges are not allowed. Assume that **no vertex is isolated** (i.e., every vertex has degree at least 1).
(a) How many such graphs have **exactly one connected component** (i.e., the graph is **connected**)?**(b)** How many such graphs have **exactly two connected components**?

2.(10%) A network engineer is designing a **fault-tolerant routing backbone** for a data-center fabric. The backbone consists of **20 routers**, arranged physically in a **4×5 cylindrical grid**: There are 4 horizontal layers ("rings"), each containing 5 routers. Horizontally, each layer forms a **5-router cycle** (i.e., the leftmost and rightmost routers are connected). Vertically, each of the 5 columns also forms a **4-router cycle** (i.e., the top and bottom switches are connected). Thus, every router has degree 4: two connections within its horizontal ring, and two within its vertical ring. **Determine whether** this cylindrical multi-ring backbone H **can be embedded on a flat PCB without wiring intersections. Justify your answer using only Euler's formula and girth-based edge bounds.** In graph theory, the girth of an undirected graph is the length of the shortest cycle contained in the graph

3.(10%) A self-organizing **ad-hoc satellite network** consists of multiple satellites placed at arbitrary integer coordinates in 3-dimensional space. Each satellite is modeled as a point $A_i \in \mathbb{Z}^3$. A **secure communication link** may be established between satellites A_i and A_j only if the **midpoint** of the segment A_iA_j also lies on an integer lattice point. Equivalently, all three coordinate-wise averages must be integers:

$$\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2}, \frac{z_i + z_j}{2} \in \mathbb{Z}.$$

The Space Agency wants to guarantee that even without knowing the positions of the satellites beforehand, **there must exist at least one secure link.**

(a) Determine the **smallest integer** N such that: No matter how the N satellites are placed in $\mathbb{Z}^3$, **there is always at least one pair** whose midpoint is a lattice point (i.e., at least one secure link always exists).

(b) Now modify the secure-link condition:
A link is allowed if **the total coordinate sum of the two satellites is even,** i.e.

$$x_i + x_j + y_i + y_j + z_i + z_j \equiv 0 \pmod 2.$$

Under this new rule, determine the **new minimum number of satellites** guaranteeing that at least **one secure link** must exist.

4.(10%) Consider the recurrence: $E_n = 4E_{n-1} - 4E_{n-2} + 2^n + 3^n$, with initial values, $E_0 = 2, E_1 = 10$. Estimate the **smallest** n such that $E_n > 10^9$.

5. (10%) You are designing the firmware logic for a high-security electronic lock. The lock receives a stream of binary signals (0 or 1) from a keypad. You need to **draw the state transition diagrams for two different versions of the security protocol. Use Moore Machine model for both versions.** Specifically, use circles to represent states with labels Si where i is the index starting from 0, use arrows to represent transitions, use an arrow to point to the start state, and mark the "Unlock" state as double circles.

(a) The Standard "Header-Trailer" Protocol Specification

1. Start-up Check: When the system first turns on, it must check for a specific "Header." The very first two signals received must be a 1 followed by another 1.
2. Security Violation: If the first two signals are anything other than a pair of 1s, the system detects a hack. It must immediately enter a Permanent Lockout state where it ignores all future input and never unlocks.
3. Monitoring: Once the valid header (11) is received, the system waits. It ignores the specific pattern of the signals in the middle.
4. Unlocking: The door should unlock if and only if the last two signals received were a 0 followed by another 0.
5. Relocking: If the system receives more signals after unlocking that break the "00" ending, the door must lock again and go back to monitoring for a new "00" ending.

(b) The Advanced "Strict Rhythm" Protocol Specification

We request a security update to the logic designed in **(a)**. Please modify your Finite State Machine to incorporate the following three specific changes. All other behaviors (like the "11" start-up check) remain the same.

1. Enforce Alternating Data: Instead of ignoring the signals between the header and the unlock command, the system must now strictly enforce an alternating "0, 1, 0, 1..." pattern.
2. Immediate Lockout on Error: If this alternating rhythm is broken (for example, if two 1s appear consecutively after the header), the system must immediately enter the Permanent Lockout state.
3. One-Time Access: The "Relock" feature is removed. Once the door unlocks (upon receiving the "00" suffix), any further input signals are considered a security violation and must trigger the Permanent Lockout state.

二. Linear Algebra (50%)

6. By factoring $\mathbf{A}$ into $\mathbf{LU}$, the LU decomposition allows solving $\mathbf{Ax} = \mathbf{b}$ by first solving $\mathbf{Ly} = \mathbf{b}$, followed by $\mathbf{Ux} = \mathbf{y}$, each requiring only a single triangular solve. Thus, computing $\mathbf{A} = \mathbf{LU}$ is crucial in large-scale computer engineering applications.

   (a) Give the complexity of the classic LU decomposition, shown in Algorithm 1. (10%)

Algorithm 1 LU Decomposition Without Pivoting

Require: Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$

Ensure: Lower triangular matrix $\mathbf{L}$, Upper triangular matrix $\mathbf{U}$ such that $\mathbf{A} = \mathbf{LU}$

Initialize $\mathbf{L} \leftarrow \mathbf{I}_n$, $\mathbf{U} \leftarrow \mathbf{0}_{n\times n}$
for $k = 1$ to n **do**
    for $j = k$ to n **do** ▷ Compute k-th row of $\mathbf{U}$
        $\mathbf{U}_{k,j} \leftarrow \mathbf{A}_{k,j} - \sum_{s=1}^{k-1} \mathbf{L}_{k,s} \cdot \mathbf{U}_{s,j}$
    end for
    for $i = k+1$ to n **do** ▷ Compute k-th column of $\mathbf{L}$
        $\mathbf{L}_{i,k} \leftarrow \frac{1}{\mathbf{U}_{k,k}} \left(\mathbf{A}_{i,k} - \sum_{s=1}^{k-1} \mathbf{L}_{i,s} \cdot \mathbf{U}_{s,k}\right)$
    end for
end for
return $(\mathbf{L}, \mathbf{U})$

   (b) An improvement of (a) based on the block matrix multiplication is shown in Algorithm 2. Give the complexity of the speed-improved LU decomposition. (10%)

Algorithm 2 Speed-Improved Block LU Decomposition

Require: Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, n is a power of 2

Ensure: Matrices $\mathbf{L}, \mathbf{U}$ with $\mathbf{A} = \mathbf{LU}$

function SpeedImprovedBlockLU($\mathbf{A}$)
    $n \leftarrow$ size of $\mathbf{A}$
    if $n = 1$ **then**
        return $([1], [\mathbf{A}_{11}])$
    end if
    Partition $\mathbf{A}$ into blocks: $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$
    $(\mathbf{L}_{11}, \mathbf{U}_{11}) \leftarrow$ SpeedImprovedBlockLU($\mathbf{A}_{11}$)
    $\mathbf{U}_{12} \leftarrow$ SISolve($\mathbf{L}_{11}, \mathbf{A}_{12}$, lower)
    $\mathbf{L}_{21} \leftarrow$ SISolve($\mathbf{U}_{11}, \mathbf{A}_{21}$, upper)
    $\mathbf{S} \leftarrow \mathbf{A}_{22} -$ SIMultiply($\mathbf{L}_{21}, \mathbf{U}_{12}$)
    $(\mathbf{L}_{22}, \mathbf{U}_{22}) \leftarrow$ SpeedImprovedBlockLU($\mathbf{S}$)
    Construct: $\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix}$, $\mathbf{U} = \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ 0 & \mathbf{U}_{22} \end{bmatrix}$
    return $(\mathbf{L}, \mathbf{U})$
end function

```
function SISolve(T, B, type)
    if n = 1 then
        return B/T
    end if
    Partition T, B into blocks:

        T = [T11  T12] , B = [B1]
            [T21  T22]       [B2]

    if type = lower then
        X1 ← SISolve(T11, B1)
        X2 ← SISolve(T22, B2 − T21 X1)
    else
        X2 ← SISolve(T22, B2)
        X1 ← SISolve(T11, B1 − T12 X2)
    end if
    return [X1]
           [X2]
end function
```

```
function SIMultiply(A, B)
    if n = 1 then
        return A · B
    end if
    Partition A, B into:

        A = [A11  A12] , B = [B11  B12]
            [A21  A22]       [B21  B22]

    M1 ← (A11 + A22)(B11 + B22)
    M2 ← (A21 + A22)B11
    M3 ← A11(B12 − B22)
    M4 ← A22(B21 − B11)
    M5 ← (A11 + A12)B22
    M6 ← (A21 − A11)(B11 + B12)
    M7 ← (A12 − A22)(B21 + B22)
    C11 ← M1 + M4 − M5 + M7
    C12 ← M3 + M5
    C21 ← M2 + M4
    C22 ← M1 − M2 + M3 + M6
    C ← [C11  C12]
        [C21  C22]
    return C
end function
```

(c) Given $A = \begin{bmatrix} 4 & 3 \\ 6 & 3 \end{bmatrix}$, compute the LU decomposition $A = LU$ using the speed-improved LU decomposition method in (b). (10%)
(Note: List the key intermediate results from Algorithm 2 to receive full credit.)

7. Let $A \in \mathbb{R}^{m\times n}$ with $m > n$, and assume that A is rank-deficient, i.e., $\text{rank}(A) < n$. $A^\top A$ is positive semi-definite.

   (a) Let $\lambda > 0$. Show that the matrix $A^\top A + \lambda I$ is positive definite. (10%)

   (b) Derive the closed-form solution for

$$x^* = \arg\min_{x\in\mathbb{R}^n} \|Ax - b\|^2 + \lambda\|x\|^2, \quad \lambda > 0. \qquad (10\%)$$