國立臺灣師範大學 115 學年度碩士班招生考試試題

科目：軟體基礎　　適用系所：資訊工程學系

注意：1.本試題共 6 頁，請依序在答案卷上作答，並標明題號，不必抄題。2.答案必須寫在指定作答區內，否則依規定扣分。

Question 1 (10 points)

Algorithm 1 is the pseudo-code for the forward pass of a 1-D convolution neural network (CNN) layer with a kernel size of k, stride of 1, dilation rate of 1, and no bias term. Denote the input features as $X \in \mathbb{R}^{C_{in} \times L}$, the output features as $Y \in \mathbb{R}^{C_{out} \times (L-k+1)}$. Assume that $k << L$, express the worst-case runtime of this function in big-Θ notation as a function of C_{in}, C_{out}, k, L.

Algorithm 1 1D Convolution

Input: $X \in \mathbb{R}^{C_{in} \times L}$
Parameters: $W \in \mathbb{R}^{C_{out} \times C_{in} \times k}$
Output: $Y \in \mathbb{R}^{C_{out} \times (L-k+1)}$
for $i = 0$ **to** $C_{out} - 1$ **do**
  for $j = 0$ **to** $L - k$ **do**
    $sum_val \leftarrow 0$
    for $c = 0$ **to** $C_{in} - 1$ **do**
      for $m = 0$ **to** $k - 1$ **do**
        $sum_val \leftarrow sum_val + X[c, j + m] \times W[i, c, m]$
      end for
    end for
    $Y[i, j] \leftarrow sum_val$
  end for
end for
return Y

Question 2 (10 points)

We construct an undirected graph G with $|V|$ vertices as follows. For each pair of vertices u, v, we randomly decide whether (u, v) will be included in E, which is the set of edges, with a chance of 50%. We repeat this process for all possible pairs of vertices (including the case when $u == v$). Express the expected value of edges in terms of $|V|$.

Question 3 **(10 points)**

The pseudo-code of the Rabin-Karp algorithm is shown in Algorithm 2. Assume that $P = 12$ and $r = 10$, i.e., we use the radix-10 system to represent a string that consists of only digits $(0, 1, 2, ..., 9)$, and $q = 11$. Find a T with exactly 8 digits such that spurious hit occurs for all $T[i..i+1]$, where $0 \leq i < \text{length}(T) - 1$.

Algorithm 2 The Rabin-Karp algorithm

```
Input: Text T, Pattern P, Radix d, Prime q
Output: All indices where P occurs in T
n ← length(T); m ← length(P)
h ← d^(m−1) mod q
p ← 0; t ← 0
for i = 0 to m − 1 do
  p ← (d · p + P[i]) mod q
  t ← (d · t + T[i]) mod q
end for
for s = 0 to n − m do
  if p = t then
    if P[0...m − 1] = T[s...s + m − 1] then
      print "Pattern occurs at shift" s
    else
      print "Spurious hit at shift" s
    end if
  end if
  if s < n − m then
    t ← (d(t − T[s] · h) + T[s + m]) mod q
  end if
end for
```

Question 4 **(10 points)**

Priority queue stores a set of n elements with their respective keys. Using a max-heap, we can achieve a worst-case time complexity of $O(\log n)$ for MAX-HEAP-INSERT (insert an element into the priority queue) and HEAP-EXTRACT-MAX (remove the element with the largest key from the priority queue) operation. Show how to build a modified version of priority queue that supports MAX-HEAP-INSERT and a new function, HEAP-EXTRACT-FIRST-QUARTILE, which returns the $\lceil n/4 \rceil$-th largest element, where n is the current number of elements in the priority queue. Both functions should have a worst-case runtime of $O(\log n)$.

Question 5 **(10 points)**

The idea of *tree sort* is to first build a binary search tree (BST) and then perform in-order traversal, which naturally prints nodes in non-decreasing order. The pseudo-code of tree sort is shown in Algorithm 3. Assume that we implement tree sort using an unbalanced BST (with **no** self-balanced mechanism). Express the best-case and worst-case runtime of the algorithm in big-Θ notation and justify your answer by providing one example for best-case and one for worst-case.

Algorithm 3 The *tree sort* algorithm.

Input: Array A of size n
Target: Sorted elements printed in order
$root \leftarrow$ Node($A[1]$)
for $i = 2$ **to** n **do**
  $root \leftarrow$ INSERT_NODE($root, A[i]$)
end for
procedure INORDER($node$)
if $node \neq$ NULL **then**
  INORDER($node.left$)
  print $node.val$
  INORDER($node.right$)
end if
end procedure
INORDER($root$)

Question 6 **(10 points)**

Let $f(n)$ and $g(n)$ be asymptotically positive functions. Which of the following conjecture(s) is(are) true? This is a multiple-choice question, and you'll earn the point once your answer is exactly correct.

(A) $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.

(B) $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$.

(C) $f(n) + o(f(n)) = \Theta(f(n))$.

(D) $f(n) = \Theta(f(n/2))$.

(E) $f(n) = O((f(n))^2)$.

Question 7 **(10 points)**

Let $R(i,j)$ be the number of times that table entry $m[i,j]$ is referenced while computing other table entries in a call of MATRIX-CHAIN-ORDER (see Figure 1). Please calculate the total number of references for the entire table. In other words, what is the value of $\sum_{i=1}^{n}\sum_{j=i}^{n} R(i,j)$ in terms of n?

```
MATRIX-CHAIN-ORDER(p, n)
 1  let m[1:n, 1:n] and s[1:n-1, 2:n] be new tables
 2  for i = 1 to n
 3      m[i, i] = 0
 4  for l = 2 to n
 5      for i = 1 to n - l + 1
 6          j = i + l - 1
 7          m[i, j] = ∞
 8          for k = i to j - 1
 9              q = m[i, k] + m[k + 1, j] + p_{i-1} p_k p_j
10              if q < m[i, j]
11                  m[i, j] = q
12                  s[i, j] = k
13  return m and s
```

Figure 1: Pseudocode of Matrix-Chain-Order.

Question 8 **(10 points)**

The edges can be classified into four types: tree, back, forward, and cross edges after performing a depth-first search. While depth-first search is applied to the graph in Figure 2 (choose node 0 as source), how many tree edges and back edges are present in this graph? Please traverse the graph in ascending order.

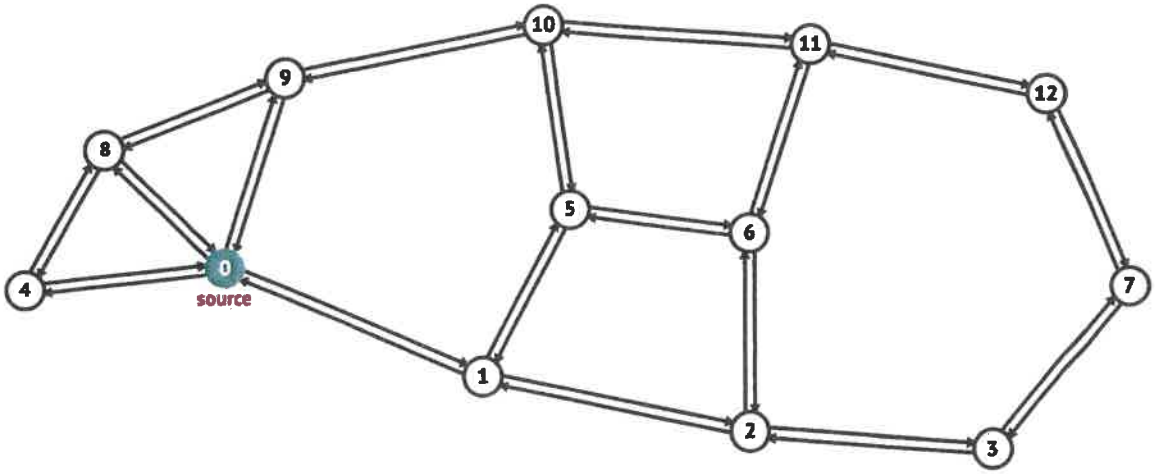


Figure 2: A directed graph for use in Question 8.

Question 9 **(10 points)**

Apply the Floyd-Warshall algorithm to the graph shown in Figure 3 and update the shortest path distances table until such an algorithm is terminated. Please use the same format as in Table 1 on the answer sheet.

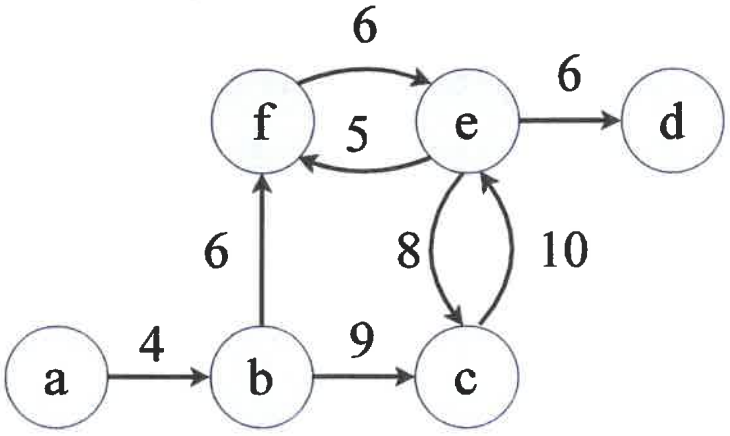


Figure 3: An undirected graph for use in Question 9.

	a	b	c	d	e	f
a	0	∞	∞	∞	∞	∞
b	∞	0	∞	∞	∞	∞
c	∞	∞	0	∞	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	∞	0	∞
f	∞	∞	∞	∞	∞	0

Table 1: Initial state of all-pairs shortest paths for the example presented in Figure 3.

Question 10 **(10 points)**

Huffman code is an efficient algorithm for lossless data compression that assigns variable-length binary codes to symbols based on their frequency, giving shorter codes to frequent symbols and longer codes to rare ones, thus minimizing overall file size.
The results of the 2025 Taiwan Annual Representative Character Election have been announced. The word "recall" was elected with a high vote of 15,084, becoming the most representative character of the year and reflecting the strong shock caused by the mass recall issue in society. Let's examine the different forms of this important word. What is an optimal Huffman code for the following context?

MMMMMMAAAASSSSSRRRRRRRRREECCCCCCCCCL

Note that lower frequencies are always on the left leaf of the binary tree and encoded as 0.

1. What is the Huffman code for the character A and S?

   (A) A : 011, S :100
   (B) A : 001, S :110
   (C) A : 101, S :010
   (D) A : 001, S :101

2. What is the plain text that decodes the "101101001001010010000" according to the Huffman code?

   (A) RSRCACAL
   (B) MASSRACE
   (C) SMALLREC
   (D) LLEACASE