1. (4 points) Let S be a stack containing $n+1$ coefficients $a_0, a_1, \ldots, a_n$, where $n \geq 1$, a_n is at the top of the stack, and a_0 is at the bottom. The following algorithm computes $f(x) = \sum_{i=0}^{n} a_i x^i$. Identify the missing expression $\square$ in the pseudocode.

```
POLY(S, x)
1  while S.size ≥ 2
2      u = S.pop()
3      v = S.pop()
4      S.push(□)
5  return S.pop()
```

(A) $v \cdot x + u$

(B) $u \cdot x + v$

(C) $v \cdot u + x$

(D) $u \cdot u + v \cdot x$

(E) None of the other choices.

2. (4 points) For positive functions $f(n)$ and $g(n)$ on $\mathbb{N}$. If $f(n) = O(g(n))$ and $f(n) = \Omega(1)$, how many of the following statements are true?

- $g(n) = \Omega(f(n))$
- $f(n) \cdot \log(1 + f(n)) = O(g(n) \cdot \log(1 + g(n)))$
- $2^{f(n)} = O(2^{g(n)})$
- $(f(n))^k = O((g(n))^k)$ for any $0 < k < 1$

(A) 0

(B) 1

(C) 2

(D) 3

(E) 4

3. (4 points) In 2022, DeepMind's AlphaTensor discovered a new state-of-the-art strategy for multiplying two 4×4 matrices using only 47 multiplications (under a special arithmetic) instead of the original 49. By integrating this method into a standard divide-and-conquer framework, the complexity of multiplying two $n \times n$ matrices is reduced to $O(n^k)$. Compute k based on the above information to get the tightest upper bound.

(A) $\log_2 47$

(B) $\log_4(47 + 49)$

(C) $\log_4 47$

(D) $\log_{16}(47 + 49)$

(E) $\log_{16} 47$

國立臺灣大學 115 學年度碩士班招生考試試題

4. (4 points) Which of the following □ generates a uniformly random permutation of an array A of size n in place, assuming that $\textsc{Random}(\ell, r)$ returns an integer chosen uniformly at random from $\{\ell, \ell+1, \ldots, r-1, r\}$?

```
RANDOM-PERMUTE(A)
for i = 1 to n
    □
    if a ≤ b
        swap A[i] with A[RANDOM(a, b)]
```

(A) $a = \textsc{Random}(1, n), b = \textsc{Random}(a, n)$

(B) $a = 1, b = i - 1$

(C) $a = i, b = n$

(D) $a = i + 1, b = n$

(E) $a = 1, b = n$

5. (4 points) Which of the following statements is known to be true?

(A) Every NP-hard problem is polynomial-time reducible to every problem in NP.

(B) Every NP-complete problem is polynomial-time reducible to every problem in NP.

(C) Every problem solvable in polynomial time is polynomial-time reducible to every problem in NP.

(D) Every problem solvable in exponential time is polynomial-time reducible to every problem in NP.

(E) None of the other choices.

6. (4 points) How many of the statements below are known to be true?

- Every problem in NP is polynomial-time reducible to some NP-hard problem.
- Every problem in NP is polynomial-time reducible to some NP-complete problem.
- Every problem in NP is polynomial-time reducible to some problem solvable within polynomial time.
- Every problem in NP is polynomial-time reducible to some problem solvable within exponential time.

(A) 0

(B) 1

(C) 2

(D) 3

(E) 4

7. (4 points) A company needs to assign **3 workers** to **3 tasks**. Each worker can perform only the tasks listed below:

- Worker W_1: Tasks T_1, T_2
- Worker W_2: Task T_1
- Worker W_3: Tasks T_2, T_3

Each worker can be assigned to at most one task, and each task to at most one worker. Which of the following statements is true?

(A) The maximum matching has size 2.

(B) The maximum matching has size 1.

(C) No matching exists.

(D) There are exactly two distinct maximum matchings.

(E) There exists a perfect matching.

接次頁

8. (4 points) Given the string $S =$ wuwvwuuv, compute the Knuth-Morris-Pratt failure function $\pi[0 \dots 7]$, where $\pi[i]$ is the length of the longest proper prefix of $S[0 \dots i]$ which is also a suffix of $S[0 \dots i]$. Which of the following is correct?

   (A) $\pi = [0, 0, 1, 0, 1, 2, 0, 0]$
   (B) $\pi = [0, 0, 1, 0, 1, 2, 2, 3]$
   (C) $\pi = [0, 1, 0, 1, 2, 0, 1, 2]$
   (D) $\pi = [0, 0, 0, 1, 1, 2, 3, 3]$
   (E) $\pi = [0, 0, 1, 1, 2, 2, 2, 3]$

9. (4 points) Which property of the root of unity $W_n = e^{2\pi i/n}$, where n is a power of 2, allows the Cooley-Tukey fast-fourier transform algorithm to achieve its speedup in time complexity?

   (A) **The Magnitude Property:** $|W_n^k| = 1$ for all k.
   (B) **The Symmetry Property:** $W_n^{k+n/2} = -W_n^k$, allowing dividing the problem into even/odd subproblems.
   (C) **The Orthogonality Property:** The sum of all roots $\sum_{k=0}^{N-1} W_n^k = 0$.
   (D) **The Conjugate Property:** $W_n^{n-k} = \overline{W_n^k}$.
   (E) **The Count Property:** There are exactly n unique roots W_n^k across all k.

10. (4 points) Consider an array [4, 5, 1, 10, 2, 7]. What does the array look like after BUILD-MAX-HEAP (bottom-up heapify) is performed, a process that converts the array into a max-heap by adjusting subtrees starting from the bottom non-leaf nodes and working up to the root?

   (A) [10, 5, 7, 4, 2, 1]
   (B) [10, 7, 5, 4, 2, 1]
   (C) [1, 2, 4, 5, 7, 10]
   (D) [10, 7, 1, 5, 2, 4]
   (E) None of the other choices.

11. (4 points) Which pivot selection strategy in Quicksort is usually the most effective at avoiding the $O(n^2)$ worst-case runtime across arbitrary input distributions?

   (A) Always select the first element.
   (B) Always select the maximum element.
   (C) Select a random element.
   (D) Select the maximum of the first and last elements.
   (E) Pivot selection does not change the algorithm's execution time.

12. (4 points) Bucket sort has an expected linear runtime, $O(n)$, under which of the following fundamental conditions?

   (A) The input values are uniformly distributed over the interval.
   (B) The input array is already sorted.
   (C) All elements are placed in a single bucket.
   (D) The number of buckets is the same as the number of input elements.
   (E) The number of buckets is smaller than the number of input elements.

13. (4 points) Which algorithm guarantees finding the median (the $n/2$-th order statistic) of an array of n elements in $O(n)$ worst-case time?

   (A) Sorting the array and then picking the median.
   (B) Quickselect using random pivots.
   (C) Using a max-heap or min-heap to extract the median.
   (D) The median-of-medians selection algorithm.
   (E) None of the other choices.

14. (4 points) A queue Q is implemented using two stacks S_{in} and S_{out} as follows. If each PUSH or POP with the stack has a cost of $\Theta(1)$, what is the amortized cost per operation over a sequence of m legal queue operations, starting from an empty state with the first operation being ENQUEUE?

```
ENQUEUE(Q, x)
    PUSH(Q.S_in, x)

DEQUEUE(Q)
    if Q.S_out is not empty
        return POP(Q.S_out)
    while Q.S_in is not empty
        PUSH(Q.S_out, POP(Q.S_in))
    return POP(Q.S_out)
```

(A) $\Theta(1)$
(B) $\Theta(\lg m)$
(C) $\Theta(m)$
(D) $\Theta(m^2)$
(E) None of the other choices.

15. (4 points) Most hash table implementations (such as Java's `HashMap`) resize the underlying array when the load factor α exceeds a certain threshold (typically 0.75). Why is this resizing operation necessary for a hash table that uses separate chaining?

(A) **Saturation:** To prevent the table from reaching a load factor of 1.0, at which point inserting a new element would be impossible.
(B) **Ordering:** To re-sort the keys alphabetically during the transfer, enabling the use of binary search for future lookups.
(C) **Complexity:** To prevent the average length of the linked lists (collision chains) from growing linearly with the number of elements, thereby maintaining $O(1)$ average access time.
(D) **Uniqueness:** To redistribute the keys across a larger range so that all existing collisions are permanently eliminated.
(E) **Memory contiguity:** To ensure that all elements in the linked lists are stored in adjacent memory addresses, minimizing CPU cache misses and memory fragmentation.

16. (4 points) Assume that there are K possible keys $\{1, 2, \ldots, K\}$ and m hash table slots $\{0, 1, \ldots, m-1\}$ with $m > 1$. In double hashing, the probe sequence is

$$h(k, i) = \big(h_1(k) + i\, h_2(k)\big) \bmod m.$$

For a fixed key k, we want the sequence $h(k, 0), h(k, 1), \ldots$ to visit all m table slots before repeating, so that insertion succeeds whenever the table is not full. Which of the following conditions is necessary and sufficient for this property (for that key k)?

(A) m must be prime.
(B) $h_2(k)$ must be odd.
(C) $\gcd(h_2(k), m) = 1$.
(D) The first hash function h_1 must be injective (one-to-one) from $\{1, 2, \ldots, K\}$ to $\{0, 1, \ldots, m-1\}$.
(E) None of the other choices.

接次頁

17. (4 points) Consider the Binary Search Tree in Figure 1. If node 66 is removed (replacing it with its in-order successor), what is the resulting *preorder* traversal sequence?

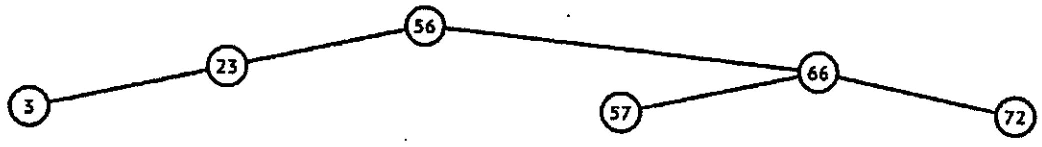

Figure 1: Binary Search Tree

(A) 3, 23, 56, 57, 72.
(B) 56, 23, 3, 72, 57.
(C) 3, 23, 57, 72, 56.
(D) 56, 23, 57, 72, 3.
(E) None of the other choices.

18. (4 points) Which of the following properties is true for *some* optimal prefix-free binary code and directly justifies the greedy choice made in the Huffman algorithm?

(A) The two symbols with the lowest frequencies must appear at the same depth in the code tree.
(B) The symbol with the lowest frequency must be assigned the longest codeword, but the symbol with the second smallest frequency need not be.
(C) All leaves at the maximum depth must correspond to symbols with the lowest frequency.
(D) The two symbols with the lowest frequencies must have equal codeword lengths, but they need not share the same parent.
(E) The two symbols with the lowest frequencies must be siblings at the maximum depth of the code tree.

19. (4 points) Consider a dynamic table that: (1) starts empty with capacity 1; (2) doubles its capacity when full; (3) shrinks its capacity by a factor of 1/2 when the capacity ≥ 4 and the number of stored elements falls below one quarter of the capacity. Assume that inserting or deleting an element costs 1, and resizing a table of size k costs $\Theta(k)$. Which statement is true when considering n insertion and removal operations?

(A) The worst-case cost of any single operation is $O(1)$, so amortized analysis is unnecessary.
(B) Using aggregate analysis, the amortized cost per operation is $\Theta(\log n)$.
(C) If shrinking were performed when the table becomes half full instead of one quarter full, the amortized cost would still remain $O(1)$.
(D) With the one-quarter shrinking rule, the amortized cost per operation is $O(1)$, but shrinking more aggressively to half capacity when the table is half-full can destroy this bound.
(E) The amortized cost depends on the order of operations and cannot be bounded independently of the operation sequence.

20. (4 points) Consider a red–black tree augmented for dynamic order statistics by storing, at each node x, the attribute

$$x.\texttt{size} = \text{number of nodes in the subtree rooted at } x, \text{including } x \text{ itself.}$$

Which of the following statements is correct?

(A) During insertion, the `size` field needs to be updated only along the path from the root to the inserted node; rotations do not require any update to `size`.
(B) During a left or right rotation, the `size` fields of exactly two nodes can be recomputed in $O(1)$ time using their children's sizes.
(C) The SELECT(i) operation (finding the i-th smallest key) requires $O(\log n)$ extra space due to recursion.
(D) Maintaining the `size` field may violate the red–black tree properties and thus requires additional recoloring rules.
(E) The presence of duplicate keys makes it impossible to support SELECT(i) correctly using subtree sizes.

21. (4 points) Let T be a B-tree of minimum degree $t \geq 2$. During the deletion of a key k from an internal node x, suppose k is stored in x at index i. Which of the following actions correctly preserves all B-tree invariants before the recursive deletion proceeds?

(A) Always replace k with its inorder predecessor from the left subtree, regardless of the number of keys in the subtrees.

(B) Replace k with either the predecessor or successor arbitrarily, since both choices always produce valid B-trees.

(C) Merge the two children of k immediately, even if both have at least t keys.

(D) Delete k directly from node x and rebalance only if an underflow occurs afterward.

(E) Replace k with the key preceeding k in the left child if the left child has at least t keys; otherwise, replace it with the key following k if the right child has at least t keys.

22. (4 points) Let $X[1 \dots m]$ and $Y[1 \dots n]$ be two sequences. Let $L[i][j]$ denote the length of the longest common subsequence (LCS) of the prefixes $X[1 \dots i]$ and $Y[1 \dots j]$, and let $C[i][j]$ denote the number of distinct LCSs (distinct sequences, not dynamic programming paths or alignments) of these prefixes. Let $C[i][0] = C[0][j] = 1$, since the empty sequence is the unique LCS in these cases. Which of the following recurrences correctly computes $C[i][j]$ without double counting identical LCS sequences?

Here, "without double counting" means that each distinct LCS sequence is counted exactly once, even if it can be obtained from multiple dynamic programming subproblems or alignments. In particular, when multiple subproblems yield LCSs of the same optimal length, any LCS sequence that appears in more than one subproblem must be counted only once.

(A) If $X[i] = Y[j]$:

$$C[i][j] = C[i-1][j-1].$$

Else if $L[i-1][j] > L[i][j-1]$:

$$C[i][j] = C[i-1][j].$$

Else if $L[i-1][j] < L[i][j-1]$:

$$C[i][j] = C[i][j-1].$$

Else:

$$C[i][j] = C[i-1][j] + C[i][j-1].$$

(B) If $X[i] = Y[j]$:

$$C[i][j] = C[i-1][j-1].$$

Else if $L[i-1][j] > L[i][j-1]$:

$$C[i][j] = C[i-1][j].$$

Else if $L[i-1][j] < L[i][j-1]$:

$$C[i][j] = C[i][j-1].$$

Else:

$$C[i][j] = C[i-1][j] + C[i][j-1] - C[i-1][j-1].$$

(C) If $X[i] = Y[j]$:

$$C[i][j] = C[i-1][j] + C[i][j-1].$$

Else:

$$C[i][j] = C[i-1][j-1].$$

(D) If $X[i] = Y[j]$:

$$C[i][j] = 1.$$

Else:

$$C[i][j] = C[i-1][j] + C[i][j-1].$$

(E) If $X[i] = Y[j]$:

$$C[i][j] = C[i-1][j-1].$$

Else:

$$C[i][j] = \max\{C[i-1][j], C[i][j-1]\}.$$

接次頁

國立臺灣大學 115 學年度碩士班招生考試試題

23. (4 points) Consider the linked-list representation of disjoint sets, where: (1) Each set is represented by a linked list; (2) Each element stores a pointer to the set's representative; (3) UNION(x, y) appends the shorter list to the longer list (union by size); (4) FIND-SET(x) runs in $O(1)$ time. Suppose we start with n singleton sets and perform any sequence of m operations consisting of UNION and FIND-SET, using union by size. Which of the following statements is correct?

    (A) The total time for all UNION operations is $\Theta(n^2)$, regardless of how unions are ordered.

    (B) The total running time of all operations is $O(m \log n)$, because FIND-SET dominates.

    (C) Using union by size makes UNION amortized $O(1)$, yielding total time $O(m)$.

    (D) The total time of all operations is $O(m+n)$, because each operation touches at most a constant number of elements.

    (E) Each element's representative pointer can be updated at most $\log n$ times, so all UNION operations take $O(n \log n)$ time in total.

24. (4 points) Let $G = (V, E)$ be a directed graph represented using adjacency lists without edge weights. A standard Breadth-First Search (BFS) is run from a source vertex s, producing distance labels $d[v]$ for all reachable vertices v. Which of the following statements is always true?

    (A) For every edge $(u, v) \in E$, we must have $d[v] \leq d[u] + 1$.

    (B) If $d[v] = d[u] + 1$, then (u, v) must be a tree edge in the BFS tree.

    (C) In a directed graph, BFS always produces a shortest-path tree that is also a topological ordering of the reachable subgraph.

    (D) If the graph is represented by an adjacency matrix instead of adjacency lists, BFS may compute incorrect distances due to repeated edge relaxations.

    (E) If BFS discovers vertices in the order $v_1, v_2, \ldots, v_k$, then this order is a shortest-path ordering for all pairs of vertices.

25. (4 points) Let $G = (V, E)$ be a connected, undirected, weighted graph with distinct and positive edge weights. Let $A \subseteq E$ be a subset of edges that is contained in some minimum spanning tree of G. Which of the following statements is always true?

    (A) Any minimum-weight edge in the graph is safe for A.

    (B) For any cut $(S, V \backslash S)$, the lightest edge crossing the cut is safe for A, regardless of whether the cut respects A.

    (C) If an edge e is the lightest edge on some cycle in G, then e must belong to every minimum spanning tree (MST).

    (D) If a cut $(S, V \backslash S)$ respects A, then the lightest edge crossing this cut is safe for A.

    (E) Any edge chosen by Prim's algorithm from an arbitrary start vertex is safe for every subset $A \subseteq E$.