高雄醫學大學 115 學年度學士後醫學系招生考試試題參考答案疑義釋疑公告

科目	題號	釋疑答覆	釋疑結果
計算機概論與程式設計	5	While the student correctly cites RFC 791 regarding the TTL field's original time-based definition, the objection stems from a misinterpretation of the mathematical concept of an "upper bound." According to the specification, every router must decrement the TTL by at least 1. Consequently, the maximum possible number of routers a datagram can traverse is exactly its initial TTL value. If processing delays cause a router to decrement the TTL by a value greater than 1, the final hop count will simply be strictly less than the initial TTL. Because the actual number of traversed routers can never exceed the initial TTL value under any circumstances, the TTL inherently constitutes a strict mathematical upper bound. Therefore, the formulation of option (C) remains logically and rigorously correct.	維持原答案
	6	While the student raises valid software engineering concerns regarding dangling pointers and correctly cites the standard O(n) LIST-DELETE operation from CLRS, the objection overlooks the explicit constraints established in the problem statement. The prompt specifically dictates deleting an element **"from the middle"** of the linked list. This condition intrinsically precludes the tail-node edge case, ensuring that a successor node always exists to facilitate the O(1) value-copying technique. Asymptotic time complexity evaluates the theoretical number of fundamental operations required, independent of system-level implementation details like memory management and external references. By copying the successor's value and bypassing the successor node, logical deletion is strictly achieved in constant time, O(1). Ultimately, the pedagogical intent of this comparative question is to contrast the localized O(1) pointer manipulation possible in a linked list with the unavoidable O(n) element shifting necessitated by contiguous memory allocation in an array. Given the stipulated premise that the target node is already known and located in the middle, option (A) remains definitively correct.	維持原答案

	11	While the student's reference to advanced literature regarding Rectified Linear Units (ReLUs) in RNNs is commendable, the objection overlooks critical technical caveats and the explicit phrasing of the question. The prompt specifically asks for the **"most effective"** architectural modification. The student's own rationale concedes that LSTM is indeed the more effective and ubiquitous solution, which directly validates option (E). The magnitude of the architectural change is not a restricting criterion in the prompt. Option (C) suggests using ReLU *exclusively*. As highlighted in the student's provided excerpt, successfully employing ReLUs in standard RNNs requires a strict prerequisite: "with the right initialization of the weights" (such as identity matrix initialization). Without this highly specific constraint, blindly replacing activation functions with ReLUs in a standard RNN typically exacerbates the *exploding* gradient problem due to the unbounded positive domain and a derivative of 1 during Backpropagation Through Time (BPTT). In contrast, the LSTM architecture intrinsically mitigates the vanishing gradient problem through its gating mechanisms and additive cell state, making it a robust and fundamentally superior structural modification without relying on fragile initialization schemes. Therefore, option (E) remains definitively correct.	維持原答案
	20	While the student demonstrates a commendable understanding of RFC 5681 and the practical edge cases of TCP congestion control, the objection conflates a standard algorithmic state transition with a global exception. In the semantics of the TCP finite state machine, a retransmission timeout acts as an interrupt. If a timeout occurs during Fast Recovery, the reduction of the congestion window (cwnd) to one segment is the direct result of the timeout event itself, not the standard termination of the Fast Recovery phase. In academic and standard protocol descriptions, "after Fast Recovery" strictly refers to the successful completion of the phase via a new ACK, at which point TCP Reno sets cwnd = ssthresh and enters Congestion Avoidance. Furthermore, the timeout scenario is explicitly and independently addressed in option (E). Forcing option (D) to encompass timeout exceptions ignores the contextual bounds of the provided choices. Pedagogically, option (D) serves as a classic distractor accurately describing the legacy behavior of TCP Tahoe, which TCP Reno was specifically designed to supersede. Thus, option (D) remains unambiguously incorrect.	維持原答案

	申論 3	While the student correctly delineates the instruction and data access proportions, the proposed calculation contains a fundamental error regarding multi-level cache architecture. The flaw lies in the treatment of the L2 cache miss rate. The student's formula adds the L2 penalty directly to the base calculation, incorrectly assuming that every executed instruction accesses the L2 cache. However, in a hierarchical cache system, the L2 cache is only accessed upon an L1 cache miss. Therefore, the provided 2% L2 miss rate is a local miss rate. To calculate the true impact on the CPI, the L2 local miss rate must be multiplied by the probability of actually accessing the L2 cache (the sum of L1 instruction and data misses): $L2_{access\_rate} = (1 * 0.05) + (0.5 * 0.10) = 0.10$ The correct L2 stall contribution per instruction is then computed using this probability: $L2_{stalls} = 0.10 * 0.02 * 50 = 0.10$ cycles By substituting this correct L2 stall contribution into the student's otherwise accurate component breakdown, the calculation mathematically resolves to an effective CPI of 1.8, rather than 2.7. Consequently, the original answer remains rigorously correct.	維持原答案