The National Higher School of Artificial Intelligence
Data Structures and Algorithms 2
Quiz # 1 19/01/2023

The quiz consists of 17 Multiple-Choice Questions.
For the first 14 questions: (1 mark for each correct answer; -0.5 for each incorrect answer)
For questions 15 to 17: (2 marks for each correct answer; -1 for each incorrect answer)

1. What data structure would you use to implement a recursive algorithm?
    A. Queue
    B. Stack
    C. Array
    D. Linked List

2. What is the worst-case time complexity of searching for an item in a linked list IF THE LIST IS NOT SORTED?
    A. O(1)
    B. O(n)
    C. O(log n)
    D. O(n log n)

3. What does '$O(n^2)$' represent in Big-O notation?
    Linear time complexity
    Exponential time complexity
    Quadratic time complexity
    Logarithmic time complexity

4. What would be the worst-case time complexity for removing a specific element **val** from a stack?
    A. O(1)
    B. O(n)
    C. O(log n)
    D. O(n log n)
    E. It depends how the stack is implemented.

5. In a singly linked list, we can add elements at both ends in...
    O(1)
    O(n)
    O(log n)
    O(n log n)

6. Linked lists are best suited?
    A. For relatively permanent collections of data
    B. When the size of the structure and the data in the structure are constantly changing
    C. For both of the above situations
    D. For none of the above situations

7. Suppose you have data sorted in ascending order in an array. What is the time complexity of inserting an element at a given position k in this list if we need to keep the data sorted?
    A. O(1)
    B. O(n)
    C. O(log n)
    D. O(n log n)
    E. The ascending order can in general not be kept after this operation

8. What would be the time complexity of finding the minimum element in a stack that would be implemented using either an array or a singly linked list?
    A. O(1)
    B. O(n)
    C. O(log n)
    D. It depends on the implementation

9. What is the time complexity of deleting a node from a singly linked list, if you only have access to that node?
    A. O(1)
    B. O(n)
    C. It is impossible to do it without access to the previous node
    D. It depends on the position of the node

10. Consider two queues implemented as linked lists with *head* and *tail* pointers. Queue-1 has n elements and Queue-2 has n elements. What is the time complexity of putting all the elements of the two queues into one queue?
    A. $\Theta(1)$
    B. $\Theta(n)$
    C. $\Theta(2\ n)$
    D. $\Theta(n^2)$
    E. $\Theta(n \log n)$

11. Which data structure can be used for efficiently tracking the smallest item in a stack of items?
    A. Another stack.
    B. A singly-linked list.
    C. An array.
    D. A doubly-linked list.

12. Considering the usual arithmetic operators priorities, what is the postfix form of the following infix expression A * B + C / D – 2 * (68 + 3) / 4 ?
    A. *A B* C D * / + 2 68 3 + * - 4 /
    B. *A B* * C D / + 2 68 - 3 4 / * +
    C. *A B* * C D / + 2 68 3 + * 4 / -
    D. *A B* * C D / + 2 68 * 3 / * 4 -
    E. Other

13. The Master theorem
    A. assumes the subproblems are of unequal sizes
    B. can be used if the subproblems are of equal size
    C. cannot be used for divide and conquer algorithms
    D. cannot be used for asymptotic complexity analysis

14. The Master theorem can be applied on which of the following recurrence relations?
    A. $T(n) = 2T(n/2) + 2^n$
    B. $T(n) = \sin(n)$
    C. $T(n) = T(n^2) + 2n^2 + 1$
    D. None of these

15. Given the recurrence relation $T(n) = T(n/2) + 2^n$, which of the following cases is true about T(n)?
    A. $T(n) = \theta(n)$
    B. $T(n) = \theta(2n)$
    C. $T(n) = \theta(n2 \log2\ n)$
    D. $T(n) = \theta(n \log2\ n)$
    E. The Master theorem does not apply

16. Given the recurrence relation $T(n) = 2T(n/2) + n/\log_2 n$, which of the following cases is true about T(n)?
    A. $T(n) = O(n)$
    B. $T(n) = O(n^2)$
    C. $T(n) = O(n^2 \log_2 n)$
    D. $T(n) = O(n \log_2 n)$
    E. The Master theorem does not apply

17. Given the recurrence relation $T(n) = 3T(n/2) + n$, which of the following cases is true about T(n)?
    A. $T(n) = \theta(n)$
    B. $T(n) = \theta(n^2 \log_2 n)$
    C. $T(n) = \theta(n^{(\log2\ 3)})$
    D. $T(n) = \theta(n^3 \log_2 n)$
    E. The Master theorem does not apply

Answers:

1-B, 2-B, 3-C, 4-B, 5-A, 6-B, 7-E, 8-B, 9-C, 10-A, 11-A, 12-C, 13-B, 14-A, 15-B (Case 3), 16-E (Case 3; Does not apply non-polynomial difference between f(n) and $n\log_b a$), 17-C (case 1)