CS2413 Assignment1 Solution

February 9, 2024

1 Time Complexity

1. Show that $n^2 + 50n = O(n^2)$, prove your answer.
   If $n^2 + 50n = O(n^2)$, $\exists c, n_0$, $\forall n > n_0$, $n^2 + 50n \leq c \times n^2$, so $1 + \frac{50}{n} \leq c$.
   As n approaches positive infinity, $\frac{50}{n}$ tends to 0, so we know that: $c \geq 1$.
   Choose $c = 2$, $1 + \frac{50}{n} \leq 2$, $n \geq 50$.
   In conclusion, $c = 2, n_0 = 50$. $\exists c, n_0$, $\forall n > n_0$, $n^2 + 50n \leq c \times n^2$,
   Therefore, $n^2 + 50n = O(n^2)$.

2. Show that $n^2 + n^2 + n^2 = 3n^2 = O(n^3)$, prove your answer.
   If $n^2 + n^2 + n^2 = 3n^2 = O(n^3)$, $\exists c, n_0$, $\forall n > n_0$, $3n^2 \leq c \times n^3$,
   so $3 \leq c \times n$, for $c > 0$, $n \geq \frac{3}{c}$
   Choose $c = 1$, $n \geq 3$.
   In conclusion, $c = 1, n_0 = 3$. $\exists c, n_0$, $\forall n > n_0$, $3n^2 \leq c \times n^3$,
   Therefore, $n^2 + n^2 + n^2 = 3n^2 = O(n^3)$.

3. The running time complexity of a linear time algorithm is given as (b).
   (a) $O(1)$; (b) $O(n)$; (c) $O(nlogn)$; (d) $O(n^2)$

4. Which notation comprises a set of all functions $h(n)$ that are greater than or equal to $cg(n)$ for all values of $n \geq n_0$? (a)
   (a) Omega notation; (b) Big O notation;
   (c) Small o notation; (d) Theta notation

5. $540n^2 + 10(=)\Omega(n^2)$

6. Which one grows faster?
   (1) $logN$ vs $\sqrt{N}$ $\sqrt{N}$
   (2) N^3 vs $1000N^2$ N^3
   (3) $log^2(N)$ vs $10log(N^5)$ $log^2(N)$

2 Linked List

1. Multiple-choice Questions
   1. A linked list is a (b)

   (a) Random access structure (b) Sequential access structure

   (c) Both (d) None of these

   2. An array is a (a)

   (a) Random access structure (b) Sequential access structure

   (c) Both (d) None of these

   3. Linked list is used to implement data structures like (d)

   (a) Stacks (b) Queues (c) Trees (d) All of these

   4. Which type of linked list contains a pointer to the next as well as the previous node in the sequence? (c)

   (a) Singly linked list (b) Circular linked list

   (c) Doubly linked list (d) All of these

   5. Which type of linked list dose not store NULL in next field? (b)

   (a) Singly linked list (b) Circular linked list

   (c) Doubly linked list (d) All of these

   6. Which type of linked list stores the address of the header node in the next field of the last node? (b)

   (a) Singly linked list (b) Circular linked list

   (c) Doubly linked list (d) Circular header linked list

2. True or False
   1. A linked list is a linear collection of data elements.(T)

   2. A linked list can grow and shrink during run time.(T)

   3. A node in a linked list can point to only one node at a time.(F)

   Tips: A node in a single linked list can only point to one node at a time.

   4. A node in a singly linked list can reference the previous node.(F)

   5. A linked list can store only integer values.(F)

   6. Linked list is a random access structure.(F)

   7. Deleting a node from a doubly linked list is easier than deleting it from a singly linked list.(T)

   Tips: Because each node in a doubly linked list has a pointer to its previous node. This means that removing a node doesn't require traversing

the entire list to find the previous node.

8. Every node in a linked list contains an integer part and a pointer.(F)

3. Fill in the Blank and Explain your answer

1. The complexity to insert a node at the beginning of the linked list is ($O(1)$).

2. Inserting a node at the beginning of the doubly linked list needs to modify (3) pointers.

Tips: The 'next' pointer of the new node is set to point to the former first node of the list. The 'prev' pointer of the new node is set to null, as there is no node before it. The 'prev' pointer of the former first node (which will become the second node after insertion) is set to point to the new node.

3. Inserting a node at the end of the circular linked list needs to modify (2) pointers.

Tips: For a singly circular linked list, we need to modify two pointers:

The 'next' pointer of the currently last node in the list, which needs to be changed from pointing to the first node to the new node. The 'next' pointer of the new node you are inserting should point to the first node of the list, maintaining the circular structure.

4. Deleting a node from the beginning of the singly linked list needs to modify (1) pointers.

Tips: The head pointer of the list will be updated to point to the second node, which becomes the new head of the list.

5. Deleting a node from the end of the circular linked list needs to modify (2) pointers.

Tips: For a singly circular linked list, we would typically need to modify at least two pointers:

The 'next' pointer of the node that is now becoming the last node in the list, which needs to point to the head of the list, maintaining the circular nature of the list. The pointer that was previously pointing to the head (now-deleted node) must be removed.

6. First node in the linked list is called the (head).

7. Overflow occurs when (there is no memory space available for new data to be stored).