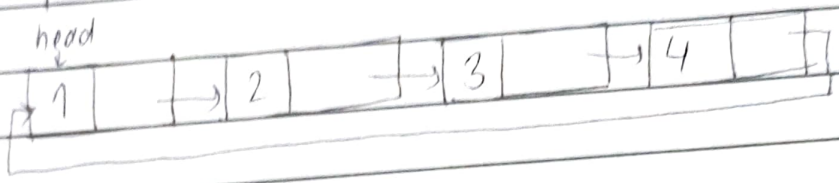
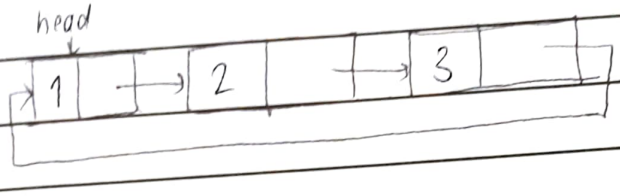
Question Nos.

Marks Awarded

Example



After deletion:



For insertion at the start, a new node was created using malloc. Newnode points to head and the last node points to the newnode. After this, we shift head to the new node. This ensures that the new node becomes the first node of the circular linked list.

For deletion of the last node, we make 2 pointers. ptr points to the last node while preptr points to the node before ptr. the last node. We use free() to delete the last node and the new last node (preptr) now points to head.

Q1] asymptotic notations are used to represent time complexity of algorithms. They are as follows:

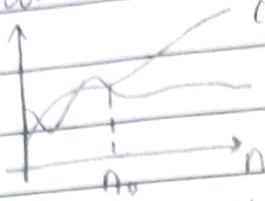
(i) Big-Oh (O) - Upper bound

This represents the worst-case time complexity. Thus, this is the highest amount of time required

Question Nos.

Marks Awarded

by the algorithm in order to execute.

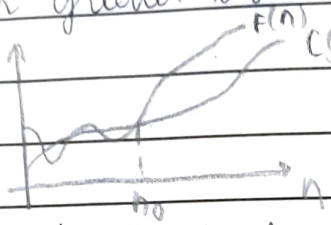


$n_0 < n$ $f(n) \leq cg(n)$ $c > 0$

Therefore, this is the worst-case scenario

(ii) Big-Omega (Ω) - Lower bound

This shows the best case time-complexity. This shows the lower bound. The time taken will be equal to or greater than this.

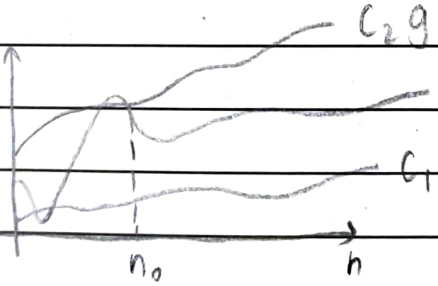


$n_0 < n$ $f(n) \geq cg(n)$ $c > 0$

Therefore, this is the best-case scenario.

(iii) Big-Theta (Θ) - Upper and lower bound.

This shows the tight bound time complexity, that is the average case. It mentions an upper and a lower bound.



$n_0 < n$, $c > 0$

$c_1g(n) \leq f(n) \leq c_2g(n)$

Therefore, this gives the average time complexity. The time complexity of the function $f(n)$ will always be between $c_1g(n)$ and $c_2g(n)$. Therefore, it gives a range for the time complexity.

These notations help us analyse the efficiency of algorithms