1. Discuss Linear vs Binary search algorithms

A:

<http://www.geeksforgeeks.org/linear-search-vs-binary-search/>

2. What is hash search? Compare it to linear search. Which is more efficient?

A:

<https://en.wikipedia.org/wiki/Hash_table>

<https://www.tutorialspoint.com/data_structures_algorithms/hash_data_structure.htm>

Searching in a hash table is not always constant-time in reality. If the hash function is a poor match for the data, you can have a lot of collisions, and in the extreme case that every data item has the same hash value, the result looks much like linear search. Depending on the details, this effective linear search might work slower than a linear search over the data in an array.

With proper and good hash function algorithm, Hash Search is efficient compared to Linear Search.

3. Write program on reversing a singly linked list

A:

static void reverse(struct Node\*\* head\_ref)

{

struct Node\* prev = NULL;

struct Node\* current = \*head\_ref;

struct Node\* next;

while (current != NULL)

{

next = current->next;

current->next = prev;

prev = current;

current = next;

}

\*head\_ref = prev;

}

4. Write a C program to sort a singly linked list

A:

<http://www.geeksforgeeks.org/c-program-bubble-sort-linked-list/>

5. Which is more efficient SLL/DLL?

A:

|  |  |  |
| --- | --- | --- |
| S.No. | Singly Linked List | Doubly Linked List |
| 1 | Singly linked list allows you to go one way direction | Doubly linked list has two way directions next and previous |
| 2 | Singly linked list uses less memory per node (one pointer) | Doubly linked list uses More memory per node than Singly Linked list (two pointers) |
| 3 | There is a little-known trick that lets you delete from a singly-linked list in O(1), but the list must be circular for it to work (move the content of next into the current, and delete next). | Doubly-linked lists can be used in places where singly-linked lists would not work (a doubly-ended queue), but they require slightly more "housekeeping", and are slightly less efficient on insertions as the result |
| 4 | Complexity of Insertion and Deletion at known position is O (n). | Complexity of Insertion and Deletion at known position is O (1). |
| 5 | If we need to save memory in need to update node values frequently and searching is not required, we can use Singly Linked list. | If we need faster performance in searching and memory is not a limitation we use Doubly Linked List |
| 6 |  | For B-Tree, Heap we need doubly linked list. .Net Framework only provides the LinkedList<T> class which is double-linked. |
| 7 | If we know in advance that element to be searched is found near the end of the list(for example name 'Yogesh' in a telephone directory), even then singly linked list is traversed sequentially from beginning. | In doubly linked list If we know in advance that element to be searched is found near the end of the list(for example name 'Yogesh' in a telephone directory), then the list can traversed from the end thereby saving time |
| 8 | In single list Each node contains at least two parts: a) info b) link | In doubly linked list Each node contains at least three parts: a) info b) link to next node c) link to previous node |

6.Write a program for traverse doubly link-lIst and sort the data and explain the approach? Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

A:

void swap (node \*x, node \*y){

int temp= x->data;

x->data=y->data;

y->data=temp;

}

void sort(node\*pointer){

int i;

while(pointer->next!=NULL){

if(pointer->data>pointer->next->data){

swap(pointer,pointer->next);

}

pointer=pointer->next;

}

sort(pointer);

}

7. There are two Singly Linked Lists with length m and n respectively; and they are intersecting each other. Write a piece of code to find node of intersection.

A: <http://www.geeksforgeeks.org/write-a-function-to-get-the-intersection-point-of-two-linked-lists/>

8. Given a pointer to node in a Doubly Linked List. Write a piece of code to delete the node.

A:

/\* Function to delete a node in a Doubly Linked List.

head\_ref --> pointer to head node pointer.

del --> pointer to node to be deleted. \*/

void deleteNode(struct Node\*\* head\_ref, struct Node\* del) {

/\* base case \*/

if (\*head\_ref == NULL || del == NULL)

return;

/\* If node to be deleted is head node \*/

if (\*head\_ref == del)

\*head\_ref = del->next;

/\* Change next only if node to be deleted is NOT

the last node \*/

if (del->next != NULL)

del->next->prev = del->prev;

/\* Change prev only if node to be deleted is NOT

the first node \*/

if (del->prev != NULL)

del->prev->next = del->next;

/\* Finally, free the memory occupied by del\*/

free(del);

}

9.Given a stack, find an element in the middle position of the stack. Optimize for space.

Implement stack using queue. Also, implement queue using stack.

A:

<http://www.geeksforgeeks.org/write-a-c-function-to-print-the-middle-of-the-linked-list/>

<http://www.geeksforgeeks.org/implement-stack-using-queue>[/](http://www.geeksforgeeks.org/implement-stack-using-queue/)

10.How many type of sortings are there and which is the best one and its complexity. Which sorting algorithm is preferred? Why? Write algorithm for best sorting techniques.

A:

<https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm>

11.Write a program to insert node after nth node using SLL

A:Only implementation Logic

if(head1 == null ){

head1 = newNode;

}

else if(position == 0){

newNode.next = head1;

head1 = newNode;

}

else{

for(int i = 0;i < position; i++){

prev = temp;

temp = temp.next;

}

prev.next = newNode;

newNode.next = temp;

}

}

12.Delete a node in a linked list program?

A: <http://www.geeksforgeeks.org/delete-a-linked-list-node-at-a-given-position/>

void deleteNode(struct Node \*\*head\_ref, int position)

{

// If linked list is empty

if (\*head\_ref == NULL)

return;

// Store head node

struct Node\* temp = \*head\_ref;

// If head needs to be removed

if (position == 0)

{

\*head\_ref = temp->next; // Change head

free(temp); // free old head

return;

}

/ Find previous node of the node to be deleted

for (int i=0; temp!=NULL && i<position-1; i++)

temp = temp->next;

// If position is more than number of ndoes

if (temp == NULL || temp->next == NULL)

return;

// Node temp->next is the node to be deleted

// Store pointer to the next of node to be deleted

struct Node \*next = temp->next->next;

// Unlink the node from linked list

free(temp->next);// Free memory

temp->next = next;// Unlink the deleted node from list

}

13.Write linked list structure and find middle node of list

A:

<http://www.geeksforgeeks.org/write-a-c-function-to-print-the-middle-of-the-linked-list/>

14/What type of memory allocation is referred for Linked lists?

A: Dynamic memory allocated from Heap

15. Write a function to reverse a singly linked list, where pointer to head is the function parameter and new head is the return value:

struct node

{

int data ;

struct node \*next;

};

struct node \* reverse\_list(struct node \* head)

{

...

...

}

A:

struct Node \*reverse(struct Node\* head\_ref)

{

struct Node\* prev = NULL;

struct Node\* current = \*head\_ref;

struct Node\* next;

while (current != NULL)

{

next = current->next;

current->next = prev;

prev = current;

current = next;

}

head\_ref = prev;

return head\_ref;

}

16.How to find a loop in singly linked list (minimum number of iterations) in optimized way? How you resolve it?

A:<http://www.geeksforgeeks.org/detect-and-remove-loop-in-a-linked-list/>

17.Difference between single/double/circular linked list

A:**Singly, Doubly & Circular Linked List**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Singly | Doubly | Circular |
| Concept | One way direction | Two way direction | One way direction in a circle |
| Has head | Yes | Yes | No-because tail will refer to first node |
| Has tail | Yes | Yes | Yes |
| No of Node | 1-next node | 2-next node & previous node | 1-next node |
| insert() | O(n) | O(1) | O(n) |
| delete() | O(n) | O(1) | O(1) |
| Benefit | require small space for each element | allow to traverse the list in both directions | execute to the end can be quickly |

18.Write a program to delete a centre node in linked list.

A: <http://www.geeksforgeeks.org/delete-middle-of-linked-list/>

// Deletes middle node and returns head of the

// modified list

struct Node\* deleteMid(struct Node \*head)

{

// Base cases

if (head == NULL)

return NULL;

if (head->next == NULL)

{

delete head;

return NULL;

}

// Initialize slow and fast pointers to reach

// middle of linked list

struct Node \*slow\_ptr = head;

struct Node \*fast\_ptr = head;

// Find the middle and previous of middle.

struct Node \*prev; // To store previous of slow\_ptr

while (fast\_ptr != NULL && fast\_ptr->next != NULL)

{

fast\_ptr = fast\_ptr->next->next;

prev = slow\_ptr;

slow\_ptr = slow\_ptr->next;

}

//Delete the middle node

prev->next = slow\_ptr->next;

delete slow\_ptr;

return head;

}

19.Write a function returning a middle node in single linked list?

A:

struct Node\* deleteMid(struct Node \*head)

{

// Base cases

if (head == NULL)

return NULL;

if (head->next == NULL)

{

delete head;

return NULL;

}

// Initialize slow and fast pointers to reach

// middle of linked list

struct Node \*slow\_ptr = head;

struct Node \*fast\_ptr = head;

// Find the middle and previous of middle.

struct Node \*prev; // To store previous of slow\_ptr

while (fast\_ptr != NULL && fast\_ptr->next != NULL)

{

fast\_ptr = fast\_ptr->next->next;

prev = slow\_ptr;

slow\_ptr = slow\_ptr->next;

}

return slow\_ptr;;

}

20.Write algorithm to swap head pointers of 2 Singly Linked Lists

A:

void swap\_headers(Node \*\* head1, Node \*\*head2)

{

Node \* temp = \*head1;

\*head1 = \* head2;

\*head2 = temp;

}

21.Write a logic for reversing doubly linked list

A:<http://www.geeksforgeeks.org/reverse-a-doubly-linked-list/>

22. Mention the steps to insert data at the starting of a singly linked list?

A:<http://codeforwin.org/2015/09/c-program-to-insert-node-at-beginning-of-singly-linked-list.html>

**Algorithm to insert node at the beginning of Singly Linked List**

**Being:**

**createSinglyLinkedList** (*head*)

**alloc** (*newNode*)

**If** (*newNode* == **NULL**) then

write ('Unable to allocate memory')

**End if**

**Else** then

read (*data*)wo

*newNode.data* ← *data*

*newNode.next* ← *head*

*head* ← *newNode*

**End else**

**End**

void insertNodeAtBeginning(int data)

{

struct node \*newNode;

newNode = (struct node\*)malloc(sizeof(struct node));

if(newNode == NULL)

{

printf("Unable to allocate memory.");

}

else

{

newNode->data = data; // Link data part

newNode->next = head; // Link address part0

head = newNode; // Make newNode as first node

printf("DATA INSERTED SUCCESSFULLY\n");

}

}

23.Mention the steps to insert data at middle of a singly linked list?

A:<http://codeforwin.org/2015/09/c-program-to-insert-node-at-middle-of-singly-linked-list.html>

**Algorithm to insert node at the middle of Singly Linked List**

**%%** Input : ***n*** position to insert data in the list

**Begin:**

**createSinglyLinkedList** (*head*)

**alloc** (*newNode*)

**If** (*newNode* == **NULL**) then

write ('Unable to allocate memory.')

**End if**

**Else** then

read (*data*)

*newNode.data* ← *data*

*temp* ← *head*

**For** *i* ← 2 to *n*-1

*temp* ← *temp.next*

**If** (*temp* == **NULL**) then

**break**

**End if**

**End for**

**If** (*temp* != **NULL**) then

*newNode.next* ← *temp.next*

*temp.next* ← *newNode*

**End if**

**End else**

**End**

24.Implement the quick sort. Compare quick sort with other sorting method and explain the differences

A: <http://www.geeksforgeeks.org/quick-sort/>

<https://www.cprogramming.com/tutorial/computersciencetheory/sortcomp.html>s

25. What is hashing technique? Describe in brief.

A: [https://s3-ap-southeast-1.amazonaws.com/tv-prod/documents/434-data%20structure%20interview%20questions%20himanshu.pdf](https://s3-ap-southeast-1.amazonaws.com/tv-prod/documents/434-data structure interview questions himanshu.pdf)

26.What is hash table? (Data Structure) Please write an example hash table code using "C" and explain

A:<https://www.tutorialspoint.com/data_structures_algorithms/hash_data_structure.htm>

27.Write a application for queue with program. Also, what do you mean by priority queue ?

A:<https://www.tutorialspoint.com/data_structures_algorithms/queue_program_in_c.htm>

28.Swapping of two linked lists with out using other linked list?

A:

29.Find if single or double linked list is circular or not

A:<http://umairsaeed.com/blog/2011/06/23/finding-the-start-of-a-loop-in-a-circular-linked-list/>

30.Mention what is the difference between Linear Array and Linked list?

A:<http://freefeast.info/difference-between/difference-between-array-and-linked-list-array-vs-linked-list/>

31.Write a program for Bubble sort

A: <http://www.geeksforgeeks.org/bubble-sort/>

32.Sort a given linked list using changing nodes.

A:<http://www.geeksforgeeks.org/merge-sort-for-linked-list/>

33.Write a code for sorting the numbers using double linked list?A:<http://www.geeksforgeeks.org/quicksort-for-linked-list/>

34.Write a program to reverse the number in singly linked list in ascending order. Write a C program to reverse a doubly linked list.

A:<http://www.geeksforgeeks.org/reverse-a-linked-list/>

<http://www.geeksforgeeks.org/reverse-a-doubly-linked-list/>

35.Delete an element in BST.

A:<http://www.algolist.net/Data_structures/Binary_search_tree/Removal>

UNANSWERED QUESTION

Given a variable n, create a single linked list of n nodes that link with circular linked list of same number of n nodes such that data in circular linked list should be multiply by 2 w. r. t single linked list of first n nodes.

Write a program to print leaf node in binary tree?

What is BFS and DFS?

Having list with some number of elements and try to add the elements at the starting without using attributes and loops?

list containing 1's and 0's and try to find the count of 1's without travelling into the list?

What is the algorithm used to store the huge data efficiently?

Binary search tree (create N nodes, traversing, display)

How to find minimum number in the list?

Describe what is Node in link list? And name the types of Linked lists?

Create a custom malloc and free function using linked lists