Pairwise swap elements of a given linked list

Given a singly linked list, write a function to swap elements pairwise. For example, if the linked list is 1->2->3->4->5 then the function should change it to 2->1->4->3->5, and if the linked list is 1->2->3->4->5->6 then the function should change it to 2->1->4->3->6->5.

METHOD 1 (Iterative)

Start from the head node and traverse the list. While traversing swap data of each node with its next node's data.

C/C++

```
/* C program to pairwise swap elements in a given linked list */
#include<stdio.h>
#include<stdlib.h>

/* A linked list node */
struct node
{
    int data;
    struct node *next;
};

/*Function to swap two integers at addresses a and b */
void swap(int *a, int *b);

/* Function to pairwise swap elements of a linked list */
void pairWiseSwap(struct node *head)
{
    struct node *temp = head;

    /* Traverse further only if there are at-least two nodes left */
    while (temp != NULL && temp->next != NULL)
    {
        /* Swap data of node with its next node's data */
        swap(&temp->data, &temp->next->data);

        /* Move temp by 2 for the next pair */
        temp = temp->next->next;
    }
}

/* UTILITY FUNCTIONS */
/* Function to swap two integers */
```

```c
void swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

/* Function to add a node at the begining of Linked List */
void push(struct node** head_ref, int new_data)
{
    /* allocate node */
    struct node* new_node =
              (struct node*) malloc(sizeof(struct node));

    /* put in the data  */
    new_node->data  = new_data;

    /* link the old list off the new node */
    new_node->next = (*head_ref);

    /* move the head to point to the new node */
    (*head_ref)    = new_node;
}

/* Function to print nodes in a given linked list */
void printList(struct node *node)
{
    while (node != NULL)
    {
        printf("%d ", node->data);
       node = node->next;
    }
}

/* Driver program to test above function */
int main()
{
    struct node *start = NULL;

    /* The constructed linked list is:
     1->2->3->4->5 */
    push(&start, 5);
    push(&start, 4);
    push(&start, 3);
    push(&start, 2);
    push(&start, 1);

    printf("Linked list before calling  pairWiseSwap()\n");
    printList(start);

    pairWiseSwap(start);

    printf("\nLinked list after calling  pairWiseSwap()\n");
    printList(start);

    return 0;
}
```

Run on IDE

Java

```
// Java program to pairwise swap elements of a linked list
class LinkedList
{
    Node head;  // head of list

    /* Linked list Node*/
    class Node
    {
        int data;
        Node next;
        Node(int d) {data = d; next = null; }
    }

    void pairWiseSwap()
    {
        Node temp = head;

        /* Traverse only till there are atleast 2 nodes left */
        while (temp != null && temp.next != null) {

            /* Swap the data */
            int k = temp.data;
            temp.data = temp.next.data;
            temp.next.data = k;
            temp = temp.next.next;
         }
    }


    /* Utility functions */

    /* Inserts a new Node at front of the list. */
    public void push(int new_data)
    {
        /* 1 & 2: Allocate the Node &
                  Put in the data*/
        Node new_node = new Node(new_data);

        /* 3. Make next of new Node as head */
        new_node.next = head;

        /* 4. Move the head to point to new Node */
        head = new_node;
    }

    /* Function to print linked list */
    void printList()
    {
        Node temp = head;
        while (temp != null)
        {
           System.out.print(temp.data+" ");
           temp = temp.next;
        }
        System.out.println();
    }

    /* Driver program to test above functions */
    public static void main(String args[])
    {
        LinkedList llist = new LinkedList();

        /* Created Linked List 1->2->3->4->5 */
        llist.push(5);
        llist.push(4);
```

```
        llist.push(3);
        llist.push(2);
        llist.push(1);

        System.out.println("Linked List before calling pairWiseSwap() ");
        llist.printList();

        llist.pairWiseSwap();

        System.out.println("Linked List after calling pairWiseSwap() ");
        llist.printList();
    }
}
/* This code is contributed by Rajat Mishra */
```

Run on IDE

Python

```
# Python program to swap the elements of linked list pairwise

# Node class
class Node:

    # Constructor to initialize the node object
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:

    # Function to initialize head
    def __init__(self):
        self.head = None

    # Function to pairwise swap elements of a linked list
    def pairwiseSwap(self):
        temp = self.head

        # There are no nodes in ilnked list
        if temp is None:
            return

        # Traverse furthur only if there are at least two
        #  left
        while(temp is not None and temp.next is not None):

            # Swap data of node with its next node's data
            temp.data, temp.next.data = temp.next.data, temp.data

            # Move temo by 2 fro the next pair
            temp = temp.next.next

    # Function to insert a new node at the beginning
    def push(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node

    # Utility function to prit the linked LinkedList
    def printList(self):
        temp = self.head
```

```
        while(temp):
            print temp.data,
            temp = temp.next


# Driver program
llist = LinkedList()
llist.push(5)
llist.push(4)
llist.push(3)
llist.push(2)
llist.push(1)

print "Linked list before calling pairWiseSwap() "
llist.printList()

llist.pairwiseSwap()

print  "\nLinked list after calling pairWiseSwap()"
llist.printList()

# This code is contributed by Nikhil Kumar Singh(nickzuck_007)
```

Run on IDE

Output:

```
Linked List before calling pairWiseSwap()
1 2 3 4 5
Linked List after calling pairWiseSwap()
2 1 4 3 5
```

Time complexity: O(n)

METHOD 2 (Recursive)

If there are 2 or more than 2 nodes in Linked List then swap the first two nodes and recursively call for rest of the list.

```
/* Recursive function to pairwise swap elements of a linked list */
void pairWiseSwap(struct node *head)
{
  /* There must be at-least two nodes in the list */
  if (head != NULL && head->next != NULL)
  {
      /* Swap the node's data with data of next node */
      swap(&head->data, &head->next->data);

      /* Call pairWiseSwap() for rest of the list */
      pairWiseSwap(head->next->next);
  }
}
```

Run on IDE

Time complexity: O(n)

The solution provided there swaps data of nodes. If data contains many fields, there will be many swap operations. See this for an implementation that changes links rather than swapping data.

Please write comments if you find any bug in above code/algorithm, or find other ways to solve the same problem.

102 Comments Category: Linked Lists

Related Posts:

- Merge two sorted linked lists such that merged list is in reverse order
- Compare two strings represented as linked lists
- Rearrange a given linked list in-place.
- Sort a linked list that is sorted alternating ascending and descending orders?
- Select a Random Node from a Singly Linked List
- Merge Sort for Doubly Linked List
- Point to next higher value node in a linked list with an arbitrary pointer
- Swap nodes in a linked list without swapping data

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.