Coding Assignment 1: CS2233

August 21, 2023

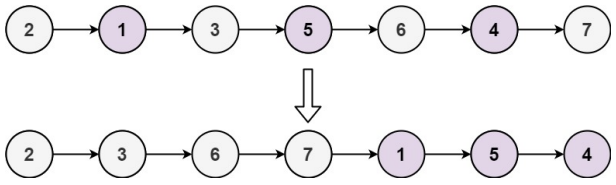
Kindly adhere to the following instructions.

- Please write a C program corresponding to each problem. Your code should be well commented and variable names should be appropriately chosen. Also prepare a `readme` text file where you can mention instructions to run the program/how to take input etc.
- Create a folder and put all the code files and `readme` text file in it, give name to the folder as "yourName_yourRollNo", zip the folder and submit it to the google classroom portal.
- Your code will also be checked against plagiarism (both from web and peer).
- Any form of plagiarism (web/chatGPT/with peers) will be severely penalised and will result in F grade.
- The submission (strict) timeline is 31st August, Thursday, 11 AM.

---

1. Given a singly linked list along with its head pointer, the problem is to group all the nodes with odd indices, followed by the nodes with even indices, and return the reordered list. The relative order inside the even and odd groups should remain in the input. The following figure illustrates the scenario:

   Your algorithm's time and space complexity should be $O(n)$ and $O(1)$, respectively.

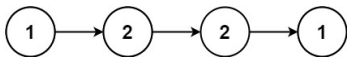


Input List = [2,1,3,5,6,4,7]
Output List = [2,3,6,7,1,5,4]

Instructions: First, write a method to create the linked list. Then, write a function that takes the head pointer of the linked list as input and performs the task. Your function can access only the linked list via the head pointer only.

(15 Marks)

2. Given a singly linked list along with its head pointer, return ***true*** if it is a *palindrome* or ***false*** otherwise. Your algorithm's time and space complexity should be $O(n)$ and $O(1)$, respectively.



Input Linked List = [1,2,2,1].
Output: ***true***.
(15 Marks)

Instructions: Same as Question 1.

3. Write a program that takes an infix arithmetic expression as input and outputs the corresponding postfix expression. You can assume that the infix statement is given in the correct format.

   For example, if the input is `a+b`, then the output should be `a b+`.
   (15 Marks)

   Instructions: The input is a string of characters, and you can assume that the input given is the correct infix format. You need to check which data structure is suitable for the purpose, and for the problem you need to use the functionality of that data structure only.

4. The dynamic-set operation SET UNION takes two disjoint sets S_1 and S_2 as input, and it returns a set $S = S_1 \cup S_2$ consisting of all the elements of S_1 and S_2. The operation usually destroys the sets S_1 and S_2. Using a suitable list data structure, show how to support SET UNION in $O(1)$ time.
   (10 Marks)

   Instructions: In this problem for simplicity you can assume that your set is the set of integers. A pair of sets S_1 and S_2 are said to be disjoint if they don't have any element in common. You need to check which data structure is suitable for the purpose, and for the problem you need to use the functionality of that data structure only. For e.g. if you plan to use stack data structure for this problem, then you can assess its elements via `Push` and `Pop` methods only. You are not allowed to use a doubly-linked

list. If you are using a singly linked list, you have access to only the head pointer of the list.

5. The dynamic-set operation SET INTERSECTION takes two overlapping sets S_1 and S_2 as input, and it returns a set $S = S_1 \cap S_2$ consisting of all the elements of S_1 and S_2. The operation usually destroys the sets S_1 and S_2. Entries of S_1 and S_2 can be in any order. Using a suitable list data structure gives the fastest possible implementation of SET INTERSECTION. (15 Marks)

   Instructions: In this problem for simplicity you can assume that your set is the set of integers. A pair of sets S_1 and S_2 are said to be overlapping if they have some element in common. You need to check which data structure is suitable for the purpose, and for the problem you need to use the functionality of that data structure only. You are not allowed to use a doubly-linked list. If you are using a singly linked list, you have access to only the head pointer of the list.

6. Write a C program to implement doubly linked lists using only one pointer value `x->np` per item instead of the usual two (`x->next` and `x->prev`). Assume that all pointer values can be interpreted as k-bit integers, and define `x->np` to be `x->np`=`x->next` XOR `x->prev`, the k-bit "exclusive-or" of `x->next` and `x->prev`. (The value NIL is represented by 0.) Be sure to describe what information you need to access the head of the list. Show how to implement the SEARCH, INSERT, and DELETE operations on such a list. Also, show how to reverse such a list in $O(1)$ time.
   (30 Marks)