Data Structures & Applications

Summer 2022

Lab 05 – Practice Tasks

**Instructor: MHM**

**Note: Keep this code with you till the course ends.**

**Task 01: (Reverse linked list)**

You have worked on all types of linkedlist, now design a method for single linked list that will reverse the linked list. Same linked list will be in reverse direction. You are given head of linked list and just have to change next pointer of nodes so that list may be reversed.

**Public Node makeReverse(Node head)**

**Task 02: (Print in reverse order)**

You are asked to design a method in linked list to print data in reverse order. You don’t need to reverse linkedlist permanently.

**Public void printReverse()**

**Task 03: (Cycle Detection):**

Write a method in linkedlist class that will detect cycle in list?

**Task 04: (Balanced Brackets)**

We have discussed in class about Balanced brackets problem using Stack. Take user string input and check whether it’s balanced or not. Use stack functions. Input may contain any of the bracket among {, [, ( and any number and letters like: ({[a+b]+c}-1) and so on.

**Task05: (FirstSingleLetter)**

Create the non-recursive function char firstSingleLetter (const char text [], const int n) which finds and returns the first letter of text that occurs only once. n is the number of characters in the text.

Here in task 5, it is not allowed to use any help structures - like for example. array, stack, queue or list. The function will therefore be as O (n2). Text contains only the letters A-Z (only uppercase letters, and no spaces).

**Task06: Apply Greedy approach**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | -1 | -1 | 1 | 1 | 1 | 0 |
| -1 | 2 | 5 | 4 | 10 | 3 | -1 |
| 3 | 2 | -1 | -1 | 0 | 3 | 8 |
| 7 | -1 | 10 | 2 | -1 | -1 | 17 |
| 4 | 3 | 9 | -1 | -1 | 8 | 33 |
| 17 | -1 | -1 | 1 | 0 | 44 | 100 |

Given a matrix above, you must start from YELLOW Box and reach to the goal which is Green Box by traversing each cell with maximum value of neighbors. -1 in any cell means BLOCK, you cannot visit that cell. You can move any cell in neighbors ONLY.

In case all the neighbors have -1 value then show message, NO POSSIBLE PATH due to BLOCKs on every side.

START: [0, 0] 🡪 YELLOW

GOAL: [n-1, n-1] 🡪 GREEN

You have to follow Greedy Approach, Greedy approach means selecting with immediate high reward (short term). In this problem, reward is your neighbors value.

Example of selecting path is given in above matrix, highlighted with ORANGE color