

Lite

PG Diploma in  
Software  
Development  
(PGDIPSWD)  
Feb 2019



Home



Discussion



Careers



NAVIGATE COURSES

## Problem Statement



PartA



Download



PartB



Download

### Part A

You have been hired to work at a new startup company called Upzilla, and Upzilla's goal is to make the best web browser for the next generation of computers. Your first project at Upzilla is to implement the browsing history feature, where you track the websites that the user has visited.

In this project, you need to implement the following functions:

**public Boolean isBrowsingHistoryEmpty()**

Returns true if the browsing history is empty

Or false if the browsing history is not empty

**public String mostRecentlyVisitedSite()**

Returns url of the website that the user most recently visited

**public void addSiteToHistory(String url)**

Adds a website url to the browsing history

**public void goBackInTime(int n)**

Removes the n most recently websites from the browsing history

Where  $1 \leq n \leq m$ , and m is the number of urls stored in the browsing

If there the user has an empty browsing history, then do nothing

**Lite**

PG Diploma in  
Software  
Development  
(Blockchain) -  
Feb 2019

**Home****Discussion****Careers**

Prints out all of the website urls in the browsing history to the command prompt

If the user has an empty browsing history, then print “no browsing history”

**Additional notes:**

isBrowsingHistoryEmpty, mostRecentlyVisitedSite, addSiteToHistory should run in  $O(1)$  time complexity

goBackInTime and printBrowsingHistory should run in  $O(N)$  time complexity

We've implemented a part of this problem for you, so you'll need to implement the rest of the solution on your own.

**Part B**

The browsing history feature was a success.

Your manager said that the users are now demanding an additional feature where the Upzilla Browser will also show websites that the users have visited the most. Therefore, your job is to implement an additional function called:

**public void listTopVisitedSites(Queue<SiteStats> sites, int n)**

sites is a queue that represents all the websites that the user has visited, as well the number of times that the user has visited each website

Please see below for definition of the SiteStats class

n represents the top n most visited sites that we want to retrieve from the queue

$0 \leq n \leq m$ , where m is the number of websites tracked by

**Lite**

PG Diploma in  
Software  
Development  
(Blockchain) -  
Feb 2019

**Home****Discussion****Careers**

The function will print the top 5 sites and the no. of times they are visited.

If multiple websites have been visited the same no. of times, then they should be ordered by recency (the last visited).

If the user has no browsing history, the function will print an empty array of type String

The SiteStats class is written as:

```
class SiteStats {
    private String url;
    private int numVisits;

    /**
     * Constructor for the SiteStats class
     *
     * @param url String that represents an URL that the user has visited
     * @param numVisits An int that represents the number of times that the user has
     */
    public SiteStats(String url, int numVisits) {
        this.url = url;
        this.numVisits = numVisits;
    }

    /**
     * This method returns the number of times that the user has visited the url.
     *
     * @return An int that represents the number of times that the user has visited t
     */
    public int getNumVisits() {
        return this.numVisits;
    }

    /**
     * This method returns the url that we are currently tracking
     *
     * @return A String that represents the url that we are currently tracking
```

## Lite

PG Diploma in  
Software  
Development  
(Blockchain) -  
Feb 2019



Home



Discussion



Careers



```

return this.url;
    }

    /**
     * This method updates the number of times that we have visited the url
     *
     * @param an int that represents the number that we want to set numVisits to
     */
    public void setNumVisits(int updatedNumVisits) {
        this.numVisits = updatedNumVisits;
    }
}

```

In addition, your manager mentions this feature will be shipped in the mobile version of Upzilla Browser. Therefore, you should use the minimal amount of space/memory possible to implement this function since memory is limited in mobile devices.

Thus, please follow these instructions while implementing the **listTopVisitedSites(Queue<SiteStats> sites, int n)** function:

Please sort Queue<SiteStats> sites without using any additional queue(s) or data structure(s)

a. In other words, sort Queue<SiteStats> sites using itself and do not allocate other data structures. You can do this by finding the most visited site and reordering the queue so that the most visited sites take the top positions after sorting is completed.

Once the queue is sorted, please print the top N sites

The runtime and space complexity of your implementation should be  $O(N^2)$  and  $O(1)$  respectively.

You would also need to complete the following method -

**public static void updateCount(String url)**

**Lite**

PG Diploma in  
Software  
Development  
(Blockchain) -  
Feb 2019

**Home****Discussion****Careers**

This method finds the given website in the queue and increments the visited count by 1, if the website is found in the queue. If the website is not found, it adds a new node to the queue.

**Note:**

In order to not limit the number of distinct sites that the users could visit from the browser, SiteStats were not stored in a constant sized array. Instead, the number of websites visited is stored in a Queue.

This also had the benefit of allowing quick updates to the numVisits variable, as user behaviour showed that they were more likely to visit sites that they had not recently gone to. Thus following a somewhat FIFO approach.

We've implemented a part of this problem for you, so you'll need to implement the rest of the solution on your own.

**Report an error****NEXT****Evaluation Rubric**