
Class Assignment 01: Practicing Basic Data Structures and Algorithms

Submission Deadline: Tuesday 9:00am, 8th November, 2016

Problem 01: Maze

In this assignment you have to implement a simple maze with hurdles and find the route from the starting point to the exit point. You need to write a program that can read a file containing the following contents:

1) 1st row of the file represents Number of Rows and Number of Columns (2D Array) separated with space.
2) 2nd row of the file represents Number of Hurdles in that particular maze.
3) Rest of the rows in the file provides Indices on which you have to mark hurdles. Each line provides you rows and columns separated with space.

The starting point of the maze is always (0, 0) and exit point will always be the last index of 2D Array. ***You do not need to hardcode the exit point since the generation of the maze is based on the input file.*** *The input file is provided to you on the course home page next to the assignment.*

Example: Figure 1 shows a simple maze consisting on 5 rows and 5 columns with 7 hurdles. The hurdle indices are (0,3), (1,4), (2,2), (3,1), (4,0), (1,1), (2,4). Starting point is (0, 0) and exit point in this case would be (4, 4). You can move in 4 possible directions only. If you move to the boundary of the maze your subject should not pass the boundary.

S			1	
	1			1
		1		1
	1			
1				E

Figure 1: Sample Maze

Program Output:

1. Your program should print the maze on the console in a similar fashion as shown in Figure 1. We do not expect you to make such a nice table on the console however you must try to print the maze easy to visualize.
2. Once the maze is created you have to print the route from starting point to the exit point**.** The route should compute a path that leads you from the starting point (row 0, column 0) to the exit **E** without observing any hurdle.

Important:

- Your program should properly allocate and de-allocate the memory.
- Try to use object-oriented approach (do not perform all the work in main).
- Do not use built in Stack, Queue, and LinkedList classes. You should create your own appropriate data structures to use.
- We would be evaluating your program by changing the input file contents so your program must be able to generate the maze accordingly.

Problem 02: Web Server Log Processing

Whenever a user browses a website, the web server hosting the website maintains a log file to record the user activity. A web server log file contains various important fields e.g., IP address of user, date time, request method, protocol, page name, response code, and response size in bytes against each request. In this assignment your objective is to read a web server log file line by line and process it to identify total number of requests and average response size against each web page. The input log file would be in the following format:

```
10.223.157.186 - - [15/Jul/2009:14:58:59 -0700] "GET / HTTP/1.1" 403 202
10.211.47.159 - - [06/Jan/2010:23:35:03 -0800] "GET /crm/campaigns_contacts.php?c=22&_search=false& 50&page=1 HTTP/1.1" 200 231
10.211.47.159 - - [06/Jan/2010:23:35:03 -0800] "GET /crm/ campaigns_contacts?c=22nd=1262849703513&rows=50 HTTP/1.1" 200 4162
10.121.241.75 - - [06/Jan/2010:23:28:59 -0800] "GET /robots.txt HTTP/1.0" 404 208
10.121.241.75 - - [06/Jan/2010:23:28:59 -0800] "GET / HTTP/1.0" 200 12758
……
……
```

If you split each line with space you will get page name on index number 6 and response size on index number 9. You need to write a program that should output the following for this input file:

```
#Page Name    Total Requests    Average Response Size
/                              2                6480
/crm/campaigns_contacts.php                     2                2196.5
/robots.txt    1                208
```

Each line in the output represents page name, total requests, and average response size separated by tab (\t).
The input file is provided to you next to the assignment at course homepage.

Important!

1. If you see a question mark (?) in page name, you need to truncate the rest of the page name. For example you can see the output for page name /crm/campaigns_contacts.php related to the sample input.
2. You may only use Map data structure from STL. You need to use your own container to store the data.
3. You need to read the input file name from command line. You may need to see a quick tutorial at http://www.cplusplus.com/articles/DEN36Up4/
4. Your program output should not contain any irrelevant output.
5. The input access log file may contain few lines that do not provide you appropriate number of tokens. Therefore your program must ignore those lines.

Submission Guidelines

You will receive an email at piazza about submission guidelines from TA's. You must comply with the submission guidelines.