CS300 – Summer 2021-2022 - Sabancı University

Homework #1 – The Traveler

Due August 6 Saturday at 23:00

Brief Description

In this homework, you will write an application that prints the coordinates of a magician salesman passes while visiting the cities where the salesman sells products. The salesman has a ritual that he follows strictly since he believes it brings good luck to his sales. He goes back to his home from the same route that he came to the last city. And your goal is to print the coordinates he traveled from leaving the house until returning back to the house.

To do this homework, you need to implement a **Stack** using whichever augmented data structure you think is appropriate. The interface of the stack class will be given, and you need to implement the function bodies as necessary. You are only allowed to modify the private section of the stack class. This is where you need to define the internal data structure which you want to use. Other than that, **YOU ARE NOT ALLOWED TO USE ANY OTHER DATA STRUCTURE!!! NOT EVEN VECTORS.**

Notice that the stack class **MUST** be a **templated** class. Using this templated class you may want to store basic data types, structs or classes in order to implement this project.

World

The world is visualized as a 2D grid and each cell is a city which the salesman could visit. These cities' coordinates are a pair of integers (x,y). The house of the salesman is at (0,0) coordinates.

The Magic of Salesman

As said, the salesman is a magician. And his superpower is, if the next city to visit was passed before, he can travel back in time to the **last** time he visited that city. It means that the visited coordinates in between the last time of a visit to that city and traveling in time back, will not be traversed by the salesmen when returning to the house because these coordinates have not been traveled already.

Stack.h given to be used in the homework assignment contains the stack class interface:

```
template <class T>
class Stack {
   private:
      // Internal data representation

   public:
      Stack();

      void push(T coor);
      void pop();

      T top();
      bool isEmpty();

      const static T noItem;
};
```

Input and Output

The coordinates of the cities which the salesman has to visit is given in a txt file. Hence, you first need to ask for a file name from the user that has the city coordinates in this file. The coordinates will be given line by line. Coordinates will be pairs of integers and they will be in the form of x and y respectively.

The output is every city that the salesman has visited and the major actions he completed. For example, while moving from home to a city, the coordinates of each city have been written. For the target city that is given as an input is written as "(x, y) is reached". "time travel back to…" is another example of a major action and the rest will be given in the demo section.

Sample Runs

Sample Run 1

input1.txt:

```
5 4

7 8

7 4

5 1
```

Output:

```
Moving to (5,4)
(1,0)
(2,0)
(3,0)
(4,0)
(5,0)
(5,1)
(5,2)
(5,3)
Arrived to (5,4)
Moving to (7,8)
(6,4)
(7,4)
(7,5)
(7,6)
(7,7)
Arrived to (7,8)
Time travel back to (7,4)
Time travel back to (5,1)
Back to (0,0)
5,0
4,0
3,0
2,0
1,0
Home sweet home
Press any key to continue . . .
```

Salesman leaves home for city (5,4) which is the first line in input. Salesman goes to that city, passing from each city on the way. This path can be an arbitrary path. So, **you can move the salesman in whatever way you wish**. After the salesman reaches (5,4), he starts moving to (7,8) which is the second input in the file. All the visited cities are also saved and printed. After he arrives at (7,8), the third input is read. Since (7,4) has been visited while coming to (7,8), now he can perform his magic! He travels in time and immediately arrives back at (7,4). The last input is read, which is (5,1) and luckily, he has visited that city as well. He can perform his magic again and goes back in time to (5,1). Since there is no input city left, he can do his ritual and goes back to his house following the route that he came to his current location. Remember that he came to (5,1) by traveling in time so all the cities visited after that were not visited at all.

Note: If the input city has been visited more than once, the salesman travels in time the last time he has visited that city.

Sample Run 2

input2.txt:

```
5 4
7 8
7 4
5 1
2 9
```

Output:

```
Moving to (5,4)
(1,0)
(2,0)
(3,0)
(4,0)
(5,0)
(5,1)
(5,2)
(5,3)
Arrived to (5,4)
Moving to (7,8)
(6,4)
(7,4)
(7,5)
(7,6)
(7,7)
Arrived to (7,8)
Time travel back to (7,4)
Time travel back to (5,1)
Moving to (2,9)
(4,1)
(3,1)
(2,1)
(2,2)
(2,3)
(2,4)
(2,5)
(2,6)
(2,7)
(2,8)
Arrived to (2,9)
Back to (0,0)
2,8
2,7
2,6
2,5
2,4
2,3
```

```
2,2
2,1
3,1
4,1
5,1
5,0
4,0
3,0
2,0
1,0
Home sweet home
```

Sample Run 3

input3.txt:

```
5 4
7 8
1 2
```

Output:

```
Moving to (5,4)
(1,0)
(2,0)
(3,0)
(4,0)
(5,0)
(5,1)
(5,2)
(5,3)
Arrived to (5,4)
Moving to (7,8)
(6,4)
(7,4)
(7,5)
(7,6)
(7,7)
Arrived to (7,8)
Moving to (1,2)
(6,8)
(5,8)
(4,8)
(3,8)
(2,8)
(1,8)
(1,7)
(1,6)
```

(1,5)
(1,4)
(1,3)
Arrived to (1,2)
Back to (0,0)
1,3
1,4
1,5
1,6
1,7
1,8
2,8
3,8
4,8
5,8
6,8
7,8
7,7
7,6
7,5
7,4
6,4
5,4
5,3
5,2
5,1
5,0
4,0
3,0
2,0
1,0
Home sweet home

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homeworks, unless otherwise noted.

How to get help?

You may ask questions to TAs (Teaching Assistants) of CS300. Office hours of TAs are at SUCourse. Recitations will partially be dedicated to clarify the issues related to homework, so it is to your benefit to attend recitations.

What and Where to Submit

Please see the detailed instructions below/in the next page. The submission steps will get natural/easy for later homeworks.

Grading and Objections

Grading:

- ❑ Late penalty is 10% off the full grade and only one late day is allowed.
- ❑ Please submit your own work only (even if it is not working). It is really easy to find "similar" programs!
- ❑ For detailed rules and course policy on plagiarism, please check out http://myweb.sabanciuniv.edu/gulsend/courses/cs201/plagiarism/

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your homework from the email address provided in the comment section of your announced homework grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Download the files you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

What and where to submit (IMPORTANT)

Submissions guidelines are below. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course).

Name your submission file:

- ❑ Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
- ❑ Name your cpp file that contains your program as follows.

  "SUCourseUserName_yourLastname_yourName_HWnumber.cpp"
- ❑ Your SUCourse user name is actually your SUNet username which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

  cago_ozbugsizkodyazaroglu_caglayan_hw1.cpp
- ❑ Do not add any other character or phrase to the file name.
- ❑ Make sure that this file is the latest version of your homework program.
- ❑ You need to submit ALL .cpp and .h files in your VS solution.

Submission:

- ❑ Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).
  1) Click on "Assignments" at CS300 SUCourse.
  2) Click Homework 1 in the assignments list.
  3) Click on the "Add Attachments" button.
  4) Click on the "Browse" button and select the zip file that you generated.
  5) Now, you have to see your zip file in the "Items to attach" list.
  6) Click on the "Continue" button.
  7) Click on the "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Resubmission:

- ❑ After submission, you will be able to take your homework back and resubmit. In order to resubmit, follow the following steps.
  1) Click on "Assignments" at CS300 SUCourse.
  2) Click Homework 1 in the assignments list.
  3) Click on the "Re-submit" button.
  4) Click on "Add/remove Attachments" button
  5) Remove the existing zip file by clicking on "remove" link. This step is very important. If you don't delete the old zip file, we get both files and the old one may be graded.
  6) Click on the "Browse" button and select the new zip file that you want to resubmit.
  7) Now, you have to see your new zip file in the "Items to attach" list.
  8) Click on the "Continue" button.
  9) Click on the "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!
Gülşen Demiröz