

CS202: Assignment 8(Recursion)

Spring 2021

1 Problem Introduction

In this assignment, you have to solve some problems using recursion. Let's talk about the problem in hand.

A group of islands is called an archipelago. In this assignment, an archipelago is made with a 2D array. An array entry of 'x' is water whereas any other character is land. An archipelago may have island. An island is formed with adjacent lands. Also, the character for land shows the color of the island and an archipelago contains islands of same color initially. You have to change the color of some islands later. Let's see few examples:

```
x r r r
x r r x
r r r x
```

This is an archipelago represented by 3*4 array and has one red island.

```
x g g x
x g x x
x x x x
x g x x
g g x x
```

This is an archipelago represented by 5*4 array and has 2 green islands.

```
x x
x x
```

This archipelago has no island. Well, technically it can't be called an archipelago as it has no island but for the sake of uniformity let's call this water body an archipelago with 0 island.

```
x b x
b x x
x x x
```

This archipelago has 2 blue islands.

So you got the idea. Now you have to perform three tasks:

First, print an archipelago recursively.

Second, count the number of islands in a given archipelago recursively.

Third, given the initial position of certain land segment in an archipelago, paint that particular island with some different color. You have to do this recursively as well.

2 Classes and Files

2.1 Archipelago class

This is a class provided to you as `archipelago.h` header file. Following are the members of the class

Archipelago
-islands : char**
-rows : int
-columns : int
-type : char
-navR: int[]
-navC: int[]
-navigate(char** , int, int): int
-isSafe(int,int) const : bool
+archipelago(char[],int,int,char);
+showIslands(int,int) const: void
+islandsCount(char **) : int
+getGrid() const: char **
+changeColor(char,int,int): void

Member variables:

- islands: This is a two dimensional array of characters to represent an archipelago.
- rows: Number of rows in an archipelago.
- columns: Number of columns in an archipelago.
- type: Color of islands in that particular archipelago.
- navR: An array initialized with {0,0,0,1,-1}. Together with navC, you can use these to navigate the archipelago. Using navR you can navigate row wise.
- navC: An array initialized with {0,1,-1,0,0}.

How to navigate using navR and navC? Suppose the current row position is row and the current column position is col. Now, to navigate:

- row+navR[0], col+navC[0]: You will be in the same position.
- row+navR[1], col+navC[1]: You will go right by one column.
- row+navR[2], col+navC[2]: You will go left by one column.
- row+navR[3], col+navC[3]: You will go down by one row.
- row+navR[4], col+navC[4]: You will go up by one row.

You can only use these to navigate. Diagonal navigation is not allowed and therefore lands adjacent diagonally are considered two islands. **Member functions:**

- archipelago(char[],int ,int, char): This is a constructor which takes 1D character array where each element is a land or water. It takes two integers for total number of rows and columns of this archipelago. Finally, it takes a char to set the color of islands. Inside the constructor, initialize a 2D array and populate it's content with each element of 1D char array passed as an argument. If an archipelago is of size 2*3, you'll get a 1D array of size 6. Set the rows and columns to the given arguments. Set the passed color value to 'type' variable. Finally, set the navR and navC with the values given above.
- islandsCount(char **): This function takes a copy of the 2D grid of your archipelago as an argument. It traverses through the grid and every time it finds a land it calls navigate function with this copy of the grid. This function returns the total number of islands. Make note, this function takes a copy of the 2D grid of your archipelago. Not the actual islands variable. This has been done so that you can change its content from 'navigate' function to mark the visited land without changing the original archipelago.

- `navigate(char **,int,int)`: This function takes a copy of the grid of an archipelago you want to navigate and a starting row and column position. That starting position always contains a land. Complete this function to visit an island. You should mark the land visited in some way so that next time you want to navigate you don't revisit the same land. Do this operation recursively. Then return the island count as 1 after each navigation.
- `isSafe(int ,int)`: This function takes two integers for current row position and column position and return true if the current position is within the boundary of an archipelago. This function has been implemented.
- `showIslands(int,int)`: This function takes starting position of an archipelago i.e. 0, 0 and prints the contents of this archipelago recursively. Complete this function.
- `getGrid()`: This function makes a deep copy of islands array to pass on to showIslands and navigate function.
- `changeColor(char,int,int)`: This function takes a new color character and two integers for row and column position. Paint the island starting from that position to the new color. This function is guaranteed to receive a row and column position of a land. Complete this function recursively.

3 Files provided

- `archipelago.h`
- `main.cpp`
- `Assignment8_instruction.pdf`(This file)
- `sample_output`

3.1 Main file

Main file contains a main function where we create some archipelago objects and return its island count, print the archipelago and change the island color. You are not allowed to change the content of this function.

4 Compiling and submitting

Compile the program:

```
g++ main.cpp -o objFile
```

Include `archipelago.h` and `main.cpp` code files in a zip file `<yourname.hw8>.zip` and submit.