# Before you start:

## Homework Files

You can download the starter files for coding as well as this *tex* file (you only need to modify *homework3.tex*) on canvas and do your homework with latex (recommended). Or you can scan your handwriting, convert to pdf file, and upload it to canvas before the due date. If you choose to write down your answers by hand, you can directly download the pdf file on canvas which provides more blank space for solution box.

## Submission Form

For homework 3, you need to upload a **tar ball** with two files in the following format:

- VE281_HW3_[Your Student ID]_[Your name].tar

  ⌊ VE281_HW3_[Your Student ID]_[Your name].pdf

  ⌊ island.cpp

Please strictly follow the format given above!!! Everyone who does not obey the format will get **2 points** deduction!!!

Notes: No extra folders (extracting this tar should only give you two files), no space in your name (use underscore(_) instead), no brackets. One example for name of tar:
**VE281_HW3_518370910000_Run_Peng.tar**

For all coding questions (question 1), you need to successfully compile your code, or otherwise you will at most get half of the whole score (we will give you partial points if you implement some of the functionalities). We will have some simple test cases to test whether your code can correctly work.

You must make sure that your code compiles successfully on a Linux operating system with g++ and the options:

```
-std=c++1z -Wconversion -Wall -Werror -Wextra -pedantic
```

Estimated time used for this homework: **3-4 hours.**

# 0  Student Info (1 point)

Your name and student id:

> **Solution:**

# 1  Millionaire! (40 points)

Roihn would like to propose to Konoka. However, she shows him a map and says: "If you wanna marry me, you need to buy all the islands on this map."

All the islands share the same price regardless of their area. In other words, Roihn only needs to know the tonal number of islands, and then pay for it. Now he is struggling with those numbers, and invites you to take the job for him. It is highly recommended to use BFS to solve this problem. The map is represented by a 0-1 2-dimension array, where 0 stands for the ocean, and 1 stands for land. Suppose the map always has the size $n \times n$, $n \in [0, 100]$. The following is one example for input.

**Sample Input:**

```
8
0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0
0 1 0 0 0 0 0 1
0 0 1 0 0 0 1 1
0 1 1 0 0 0 1 1
0 0 1 0 1 0 0 0
0 1 1 1 0 0 0 0
1 0 0 1 0 0 0 1
```

**Sample Output:**

```
6
```

**Explanation:** The first line of input is a number $n$, which indicates the size of the map.

The following $n$ lines provide the information of the map. For each land block, only those blocks on its top, bottom, left and right can be connected to itself and form one island. For this sample, there are 6 islands in total as we see in the output. They are:

```
0 0 0 0 0 0 0 0
0 a 0 0 0 0 0 0
0 a 0 0 0 0 0 d
0 0 b 0 0 0 d d
0 b b 0 0 0 d d
0 0 b 0 c 0 0 0
0 b b b 0 0 0 0
e 0 0 b 0 0 0 f
```

You can see that the island "c" is isolated, since it is not connected with island "b" according to the definition. So is the island "e".

The **output** is one number which indicates the number of islands on the given map.

Please **name your code file** as *island.cpp*. There is **NO** starter file provided for this question. Please do enough testing before submission, otherwise you may lose some points. Also, please carefully read the document on the first page and follow the given requirements.

## 2   Store Ranking (20 points)

Roihn now moves in his new home in Shanghai. Not familiar with the surrouding environment, he would like to know the information about stores nearby. Now given you one set of stores' data, which contains their locations and star ratings (range from 0 to 5). Roihn would like you to construct a **k-d tree** to maintain the stores' data, so that he can search for the most proper store efficiently according to his preference.

The following is the raw information of 10 stores:

| #  | X  | Y  | Star Rate |
|----|----|----|-----------|
| 1  | 10 | 8  | 3.5       |
| 2  | 12 | 3  | 4         |
| 3  | 6  | 7  | 3.8       |
| 4  | 32 | 11 | 2         |
| 5  | 5  | 12 | 4.8       |
| 6  | 21 | 13 | 3.4       |
| 7  | 3  | 10 | 5         |
| 8  | 4  | 27 | 4.9       |
| 9  | 1  | 1  | 3.9       |
| 10 | 13 | 9  | 2.7       |

In this table, X and Y stand for the 2D location of the store.

1. Please insert those stores' data in the given order and draw the final k-d tree. The dimension form should be $(X, Y, StarRate)$. (12 points)

   **Solution:**

2. Suppose the forth store is renewed at the same location. Because of its renew up, its star rate raises to 3. Please draw the k-d tree after the star rate is updated. (8 points)

   **Solution:**

## 3   AVL Tree (19 points)

1. Suppose that we insert a sequence of keys 2, 1, 3, 7, 9 into an initially empty AVL tree. Draw the resulting AVL tree. (3 points)

   **Solution:**

2. Suppose that we further insert key 6 into the AVL tree . Draw the resulting AVL tree. (3 points)

> **Solution:**

3. Suppose that we further insert keys 4, 5 into the AVL tree. Draw the resulting AVL tree. (3 points)

> **Solution:**

4. Suppose that we further insert keys 8, 10, 11, into the AVL tree. Draw the resulting AVL tree. (3 points)

> **Solution:**

5. Write down the balance factor for each node in the final AVL tree. (3 points)

> **Solution:**

6. Further remove the key 4 from the AVL tree. Draw the resulting AVL tree. (4 points)

> **Solution:**

# 4  RB Tree (20 points)

**Please clearly label the color of each node.**

1. Suppose that we insert a sequence of keys 9, 3, 1 into an initially empty red-black tree. Draw the resulting red-black tree. (4 points)

> **Solution:**

2. Suppose that we further insert key 6 into the red-black tree. Draw the resulting red-black tree. (4 points)

> **Solution:**

3. Suppose that we further insert keys 2, 8 into the red-black tree. Draw the resulting red-black tree. (4 points)

**Solution:**

4. Suppose that we further insert key 7 into the red-black tree. Draw the resulting red-black tree. (4 points)

**Solution:**

5. Suppose that we further insert keys 4, 5 into the red-black tree. Draw the resulting red-black tree. (4 points)

**Solution:**

# Reference

Assignment 4, VE281, FA2020, UMJI-SJTU.