Final:

Due Date: **Sunday, December 13, 2020, at 11:59pm.**

This exam contains three problems asking multiple questions. Please answer each question in detail with clear explanation. :)

Problem 1. What is the running time of the below code? **Explain in detail. Make sure to create the recursion tree and show all the calculations.**

```
Test(a, 0, a.length);    // ****************** Calling Test for the 1st time ******************

Test(a, start, end)        // Test is a function with 3 inputs. Array a, start index and end index.
    n = end - start;       // Size of part of array a.
    if n <= 1
      return a(n);
    else
      newEnd = start + n/6;
      newEnd2 = newEnd + 2*n/6;

      Sol1 = Test(a, start, newEnd);
      Sol2 = Test(a, newEnd+1, newEnd2);
      Sol3 = Test(a, newEnd2+1, end);
      CombineSol = Combine(a, start, newEnd, end); // Combine is a function with runtime of T(n) = O(n)

      return min([Sol1, Sol2, Sol3, CombineSol]);     // Return min of answers.
    end
```

Hint: Make sure to calculate all the input sizes correctly. This part is very important, so double check your answer. Read the code very carefully.

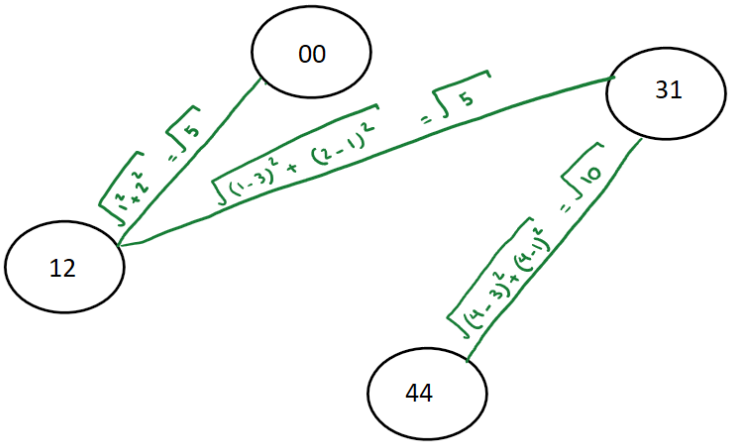
Problem 2. You are given a matrix called buildings that has the location of all the buildings at a university in a two-dimensional coordinate. We would like to construct paved paths that connect the buildings to each other. Implement an algorithm to calculate the minimum budget required to finish the constructions.

Note: We calculate the cost of connecting any two departments using the Euclidean distance. Let's say (x_i, y_i) and (x_j, y_j) are the coordinates of two buildings, the cost to connect them is:

$$cost_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

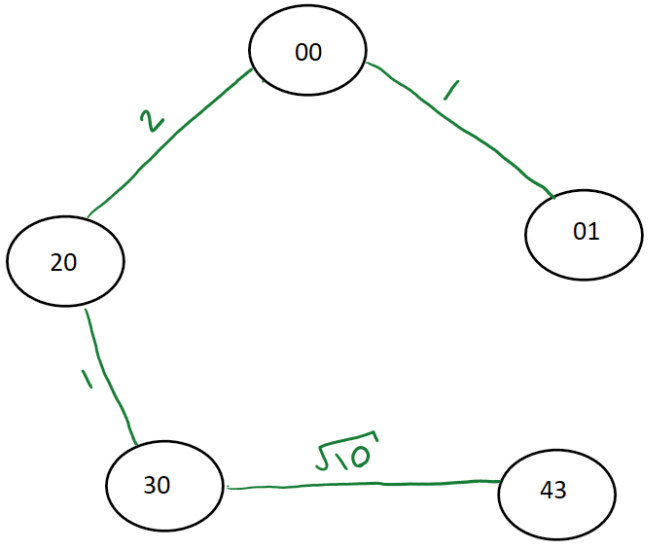
Example 1: *Input*: buildings = $\begin{bmatrix} 0 & 0 \\ 1 & 2 \\ 3 & 1 \\ 4 & 4 \end{bmatrix}$

➔ *Output*: "The minimum budget required to connect all the buildings is 7.63." $(2\sqrt{5} + \sqrt{10} = 7.63)$



Example 2: *Input*: buildings = $\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 2 & 0 \\ 3 & 0 \\ 4 & 3 \end{bmatrix}$

➔ *Output*: "The minimum budget required to connect all the buildings is 7.16." $(4 + \sqrt{10} = 7.16)$



A. How would you find the minimum amount to construct the paths? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) **Explain each solution/algorithm in detail**.

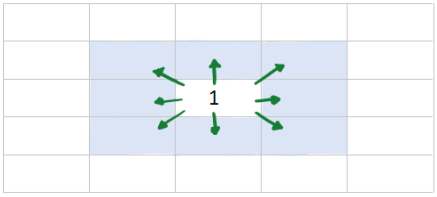
B. Write the pseudocode for the best algorithm you came up with.

C. Implement your answer using any programming language you want to.

D. What is the time complexity of your answer? **Explain in detail and show all the work**. (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)

Problem 3. You are given a matrix that has 0s and 1s in it. Implement an algorithm to find the exact number of connected components on the map.

Note: Connected component: Group of 1s that are neighbors to each other. We can have 8 possible neighbors for each elements in the middle. (Please note that the elements on the first/last row/columns have less number of neighbors)



Example 1: *Input*: m = $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$ ⟵ $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$

➔ *Output*: "The total number of connected components is 2."

Example 2: *Input*: m = $\begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$ ⟵ $\begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$

➔ *Output*: "The total number of connected components is 3."

A. How can you find the total number of connected components? (**Note**: If you have multiple answers in mind, break them apart and explain each one separately.) **Explain each solution/algorithm in detail**.

B. Write the pseudocode for the best algorithm you came up with.

C. Implement your answer using any programming language you want to.

D. What is the time complexity of your answer? **Explain in detail and show all the work**. (**Note**: If possible, break your code/pseudocode to different parts, calculate the runtime for each step and then try to calculate the total running time based on that.)