Answer the following 4 questions and submit your answers online:

1. [10 points] Analysis

There are n pancakes to be fried on a small grill that can hold only two pancakes at a time. Each pancakes has to be fried on both sides; frying one side of a pancakes takes 1 minute, regardless of whether one or two pancakes are fried at the same time. Consider the following recursive algorithm for executing this task in the minimum amount of time. If $n \leq 2$, fry the pancakes or the two pancakes together on each side. If $n > 2$, fry any two pancakes together on each side and then apply the same procedure recursively to the remaining $n - 2$ pancakes.

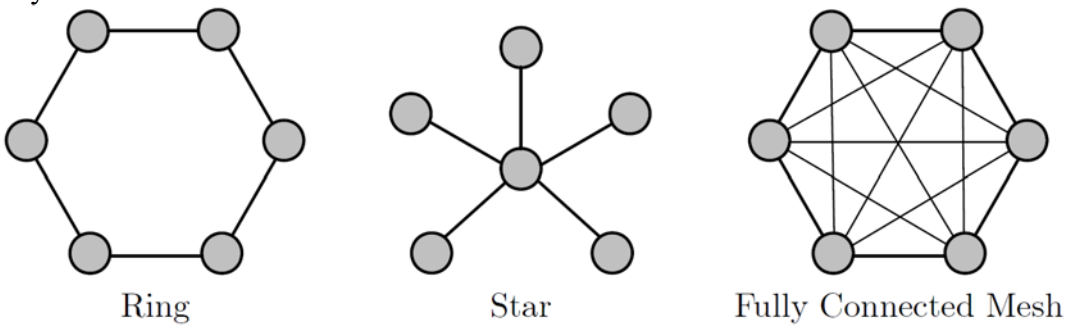
a. Set up and solve the recurrence for the amount of time this algorithm needs to fry n pancakes.

b. Explain why this algorithm does not fry the pancakes in the minimum amount of time for all $n > 0$.

c. Give a correct recursive algorithm that executes the task in the minimum amount of time.

2. [10 points] Exhaustive Search

A network topology specifies how computers, printers, and other devices are connected over a network. The figure below illustrates three common topologies of networks: Ring, Star, and Fully Connected Mesh.



Ring Star Fully Connected Mesh

You are given a boolean matrix A[0..n−1, 0..n−1], where n > 3, which is supposed to be the adjacency matrix of a graph modeling a network with one of these topologies. Your task is to determine which of these three topologies, if any, the matrix represents. Design a brute-force algorithm for this task and indicate its time efficiency class.

3. [10 points] Decrease and Conquer

a. Show that the disk moves made in the classic recursive algorithm for the Towers of Hanoi puzzle can be used for generating the binary reflected Gray code.

b. Show how the binary reflected Gray code can be used for solving the Towers of Hanoi puzzle.

4. [10 points] Divide and Conquer

How many lines does his algorithm print? Write a recurrence and solve it.

```
function printaton(n: an integer power of 2) {
    if n > 1 {
        printaton(n/2)
        printaton(n/2)
        printaton(n/2)
        for i = 1 to n ^ 4 do
            printline("are we done yet?")
    }
}
```

a. Use Master's theorem to obtain an asymptotic solution.

b. Derive an exact solution by expanding the recurrence.