CSCI201A - S4 - Divide and Conquer

Provide a recursive solution to solve the following problems using divide and conquer. Each problem will be 2 pts each. Partial points will be given for iterative solutions.

1. Squid Games Season 2

You are currently designing a game for the next Squid Games and the big boss man wants to make sure that people are eliminated but doesn't want to make it too obvious how someone could win the game. Being the crafty algorithmic wizard that you are (plus not wanting to get shot), you decided to make the game as follows:

a. Given **n** number of players, each player will be able to pick a number from 1 to 10 (assume that there are more than 10 players in the current round).
b. A player can pick a number that another player got (example, Player A picked 5 and Player B picked 5 as well).
c. Big boss man gets to choose a number **p** that is less than 10 wherein the first **p** groups with the highest count of similar numbers picked will survive.

For example, given 12 players, the resulting number picks will be as follows:

- 5 people picked 3
- 4 people picked 5
- 2 people picked 6
- 1 person picked 2

If the big boss decided to have p as 2, then the 5 people who picked 3 and the 4 people who picked 5 would survive.

Another example would be given 8 players, the resulting number picks will be as follows:

- 3 people picked 1
- 2 people picked 2
- 3 people picked 3

If the big boss decided to have p as 2, then the 3 people who picked 1 and the 3 people who picked 3 would survive.

The algorithm should be able to return what groups survived the game.

2. Optimus Prime

Write a recursive function that given a number, would return if it’s prime or not. A number is considered prime if it is only divisible by 1 and itself.

3. Optimal Prime

Given a prime number and a number n, write a function that returns the prime numbers before the given input prime number within the window of n.

For example:

23 is a prime number.

n = 5

23 - 5 = 18

From 18 to 23, there is only 1 prime number (19).

Function should return [19].

4. More Squids

A 34 year old dentist is tasked to cut off squid guts in the harbor to earn some money to pay for coffee she bought in a local cafe. Each squid that she cuts is given a certain rating from 1 to 10. At the end of the day, a surfer dude checks the container of squids and wants to determine which rating the dentist achieved the most. Write an algorithm to determine this.

For example:

squids_cut = [3, 3, 4, 4, 4, 2, 2, 1]

The algorithm should return 4.

5. Stairway to Heaven

Robert is trying to figure out the hardest path to heaven. Given a matrix of N by M, write an algorithm that determines all possible paths to heaven from the starting point (0, 0) to the ending point (lower right corner) of the matrix. A possible path to heaven is one that is in ascending order that is able to reach the end of the matrix. The function should then return the path to heaven that is the hardest; that is, the path whose sum of all elements has a maximum value.

For example:

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

The hardest path to heaven is 1+2+3+10+18+21+23+26+30 = 134. The algorithm should return [1, 2, 3, 10, 18, 21, 23, 26, 30].

For simplicity, movement can only be from left to right and top to bottom.