Designing Concurrent Algorithms

Making pizza requires three steps: preparing the dough (15 min), putting on the toppings (3 min), baking (10 min). You and two other friends are tasked to make 4 pizzas. Using pipelining (each person does exactly one task), how long will the will making 4 pizzas take? How much time do you save compared to if you made all the four pizzas alone?

Would you use MapReduce, Pipelining, or neither for the following?

Solving an equation that can be split into many steps that need to be done ins a specific order.

Running a simulation that must be run sequentially once.

Taking a large dataset of people's favorite colors, and finding how many people like purple.

What the mapper and reducer should do and return for the following problem: you have many different routes for your daily walk, and for each route you have a list of how many dogs you'll see on each block. Find which route lets you see the most dogs.

Mapper:
Reducer:

Your boss assigns your team the task of making as many powerpoint presentations as you can over the next 2 hours for your upcoming sales pitch. You and your 2 team members know that the process of making a presentation involves picking a theme (15 minutes), writing the slides (45 minutes), and making final edits (15 minutes), all in that order. Assuming you and your team members implement pipelining, what is the most number of presentations you can complete within the 2 hours? Draw a pipeline process chart supporting your answer.

Write a description for mapper, collector, and reducer functions that could be used to determine how many vowels appear in a list of strings.

In general, what information does the mapper give to the collector? The collector to the reducer? Why is MapReduce efficient?