Consider 3 different problems, solution of the heat equation in 1D with 100 points, solution of the heat equation in 1D with 10000 points, or solution of the heat equation in 2D with 1000 points. Rank them in what you would expect to be parallel efficiency with run on a GPU, assuming the most efficient possible solution.

Given a nested loop over i and j representing x and y, how would you set up your index structure in a GPU kernel (e.g. in 1D you might write `i = threadIdx.x + blockDim.x*blockIdx.x`)? Include any additional considerations you will need to make (number/dimension of blocks and threads to start the kernel, any if or while statements inside the kernel).