

Exercise Instructions for Students

1. Think of a previous computer program you have written that processes some kind of data (a list of numbers, an image, etc.). Do you think it would be possible to utilize multiple cores to parallelize it? If not, why not? If yes, how do you think you would parallelize it?
2. Why are parallel algorithms desired?
3. What do you have to watch out for when deciding if a program can be parallelized?
4. Give an embarrassingly parallel algorithm to compute a **scalar** times a **vector**. For example, $2(4, 2, -1) = (2 * 4, 2 * 2, 2 * -1) = (8, 4, -2)$.
5. In the field of *Digital Image Processing*, a digital image is stored as a matrix, where each location stores the amount of red, green, and blue at that location (3 numbers per location in the matrix). Suppose a digital image has a resolution of 1024×768 (width * height). A simple algorithm for converting the image to grayscale is to take the red, green, and blue values of each pixel, compute their mean, and store that value back as the new red, green, and blue values. Suppose 8 processes are available for parallel computation. Provide an **embarrassingly parallel algorithm** to compute the grayscale image based on this description.
6. Having an 8-core CPU usually does not mean you will have a program that runs exactly 8 times faster if it is parallelized. Why do you think this is?