

For this assignment we will develop a couple of parallel programs to run on the parallel cluster built using the raspberry pies. For each of the implemented functions you will have to add the instrumentation to measure its performance (simple time measure will do), as this will be used during the project defenses to observe the behavior of the algorithms in different hardware configurations.

Task 1. Given a (large) text file, count the occurrences of each word in the text. For the purposes of this exercise you can expect that a word will always be one of: (1) a number composed of one or more ASCII digits (*i.e.*, "0" or "1234"), (2) a simple word composed of one or more ASCII letters (*i.e.*, "a" or "they"), or (3) A contraction of two simple words joined by a single apostrophe (*i.e.*, "it's" or "they're")

When counting words you can assume the following rules: The count is case insensitive (*i.e.*, "You", "you", and "YOU" are 3 uses of the same word) (you have to make sure to count the words correctly) The count is unordered; the tests will ignore how words and counts are ordered Other than the apostrophe in a contraction all forms of punctuation are ignored The words can be separated by any form of whitespace (*i.e.*, "\t", "\n", " ")

The function specification is given by:

```
1 @doc """
2 Count the number of words in the sentence.
3
4 Words are compared case-insensitively.
5 """
6 @spec count(String.t()) :: map
```

Task 2. Image processing algorithms. We are going to implement a couple of algorithms to process images.

- Read an image (png). To read and write PNG images, you may use the Pixel library (or any other library that you see fit to do the job)
- Given an image, write a function to rotate the image by a given angle. You must implement a function `rotate(img, angle)` that receives an image, `img`, (representation) and a rotation `angle`, and returns the rotated image.
- Given two images, source and target, implement a parallel version of the Beier-Neely Method for image morphing. Your implementation should output the intermediate images of the transformation.
- Write the modified image to a PNG file

This work is to be developed in groups (two, max three people). You must hand in a project/file with the implementation of your work for each of the tasks, to run during the project defenses. Additionally, you must hand-in a report with an analysis of parallelism possibilities and analysis, as well as an analysis of the speed-up/slowdown of your solution running with different cores.