Homework 9

Task 1 :

```
The line_profiler extension is already loaded. To reload it, use:
  %reload_ext line_profiler
Timer unit: 1e-09 s

Total time: 0.032314 s
File: /tmp/ipykernel_454201/3335582310.py
Function: res_skimage at line 6

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
     6                                           def res_skimage(imgs):
     7         1       3615.0   3615.0      0.0      new_size = (imgs[0].shape[0]//2, imgs[0].shape
[1]//2)
     8         1        214.0    214.0      0.0      res_im = []
     9       201      81282.0    404.4      0.3      for im in imgs:
    10       200   32057096.0 160285.5     99.2          image_resized = transform.resize(im, new_size,
anti_aliasing=True)
    11       200      96036.0    480.2      0.3          res_im.append(image_resized)
    12         1      75738.0  75738.0      0.2      return np.asarray(res_im)
```

Task 2 :

I opted for multithreading over multiprocessing because the Pi approximation is computationally intensive and benefits from running concurrently in separate threads. In the context of the Global Interpreter Lock (GIL), which Python uses to handle threads, this task saw improvement as it involved running I/O operations alongside computations.

Task 3 :

```python
@jit(nopython=True)
def numba_optimized_approximate_pi(n):
    pi_2 = 1.0
    nom, den = 2.0, 1.0
    for i in range(n):
        pi_2 *= nom / den
        if i % 2 == 0:
            den += 2
        else:
            nom += 2
    return 2 * pi_2

nums = [1_822_725, 22_059_421, 32_374_695, 88_7
```

Task 4 :