Mahbod Afarin 862186340:

1. We have 16 warps at first. Then at step 2, 16/2 = 8 wars, step 3: 4 warps, step 4: 2 step 5: 1 warp with 32 numbers. And we continue calculating numbers in log32=5 steps. So, we don't have divergence in step1, but we have warp divergence in other steps. So, in total, 9 steps have warp divergence and 1 step doesn't have.

2. We don't have warp divergence until step 5. So, we have 5 steps with warp divergence and 5 steps without warp divergence.

3. The optimized kernel performed better than the naïve version.
In the gpgpu statistics, we can see that there are 89321 number of cycles while in the naive version, we need 135752 cycles to run the application. Thus, we can conclude that we need less cycles to do the same task in the optimized version, so the optimized version has better performance.

4. From the GPGPU statistics:

Naive Version:

W32: 2156653
W16: 176744
W8: 176744
W4: 176744
W2: 176744

Optimized Version:

W32: 2093415
W16: 8715
W8: 8715
W4: 8715
W2: 8715

So, we use less warps in the optimized version. Because we are utilizing the hardware better in this method.

5. As we had in the lecture, warps are the smallest unit of execution. In the case of having warp divergence, we are not using all threads inside a warp, so we are wasting hardware. So, we suffer from warp divergence.