Name: Hasanat Jahan
CS381-16 Assignment 4

Answer to 2.23
In this aggregation, each core would have one bin to fully increment all values of that bin. Which means that there would have to be as many cores as there are bins, which could be many cores which reduces performance significantly. Also this way, as we call Find_bin to return b for each bin, we cannot make sure to divide the work between the cores roughly equally as the number of elements in bins are different, as one bin might have much more data than other bins.

B. Describe how to install MPI and mpi4py. You may choose any MPI implementation, and any operating system. Describe all steps with commands executed and/or screenshots.
I have a MacOS so I installed MPI using homebrew. I used the command:
`brew install open-mpi`

Then I installed mpi4py using the command
`brew install mpi4py`

C. Run attached first-example.c and first-example.py using the MPI environment above. Show the output after running the two files.
Running `first-example.c`:
Terminal commands used:
```
mpicc -o first-example first-example.c
mpirun first-example
```

Output:
```
Enter the number of elements (n):
1
I am process 0 out of 8 handling the 0th part of n=1 elements,
result=0.000000
I'm proc 0: My own result is 0.000000
I am process 3 out of 8 handling the 3th part of n=1 elements,
result=3.000000
I am process 4 out of 8 handling the 4th part of n=1 elements,
result=4.000000
I am process 1 out of 8 handling the 1th part of n=1 elements,
result=1.000000
```

```
I am process 2 out of 8 handling the 2th part of n=1 elements,
result=2.000000
I am process 5 out of 8 handling the 5th part of n=1 elements,
result=5.000000
I am process 7 out of 8 handling the 7th part of n=1 elements,
result=7.000000
I am process 6 out of 8 handling the 6th part of n=1 elements,
result=6.000000
I'm proc 0: received result of process 1 is 1.000000
I'm proc 0: received result of process 2 is 2.000000
I'm proc 0: received result of process 3 is 3.000000
I'm proc 0: received result of process 4 is 4.000000
I'm proc 0: received result of process 5 is 5.000000
I'm proc 0: received result of process 6 is 6.000000
I'm proc 0: received result of process 7 is 7.000000
```

Running first-example.py:
Terminal commands used:

```
mpirun -np 5 python3 first-example.py
```

Output:

```
Enter the number of elements (n): 2
I am process 0 out of 5 handling the 0ith part of n=2
elements, result=0.0
I'm proc 0: My own result is 0.0
I am process 1 out of 5 handling the 1ith part of n=2
elements, result=2.0
I am process 2 out of 5 handling the 2ith part of n=2
elements, result=4.0
I am process 3 out of 5 handling the 3ith part of n=2
elements, result=6.0
I am process 4 out of 5 handling the 4ith part of n=2
elements, result=8.0
I'm proc 0: received result of process 1 is 2.0
I'm proc 0: received result of process 2 is 4.0
I'm proc 0: received result of process 3 is 6.0
I'm proc 0: received result of process 4 is 8.0
```