Tommy Ho
1/27/25
CS561

Project 0

C)

W1: 10K point queries on 1M sorted data*

```
[hot@scc-wi4 cs561_templatezonemaps2]$ ./workloadgenerator -N1000000 -P10000 --sort
[hot@scc-wi4 cs561_templatezonemaps2]$ ./main
Time taken to perform point queries from zonemap = 4532.33 microseconds
Time taken to perform range query from zonemap = 1.02879e+07 microseconds
```

W2: 1K range queries, selectivity: 0.001 on 1M sorted data

```
[hot@scc-wi4 cs561_templatezonemaps2]$ ./workloadgenerator -N1000000 -R1000 -s0.001
--sort
[hot@scc-wi4 cs561_templatezonemaps2]$ ./main
Time taken to perform point queries from zonemap = 4295 microseconds
Time taken to perform range query from zonemap = 1.0409e+07 microseconds
```

W3: 1K range queries, selectivity: 0.1 on 1M sorted data

```
[hot@scc-wi4 cs561_templatezonemaps2]$ ./workloadgenerator -N1000000 -R1000 -s0.1 --sort
[hot@scc-wi4 cs561_templatezonemaps2]$ ./main
Time taken to perform point queries from zonemap = 4281.33 microseconds
Time taken to perform range query from zonemap = 1.02162e+07 microseconds
```

W4: W1 with unsorted data

```
[hot@scc-wi4 cs561_templatezonemaps2]$ ./workloadgenerator -N1000000 -P10000
[hot@scc-wi4 cs561_templatezonemaps2]$ ./main
Time taken to perform point queries from zonemap = 5429.33 microseconds
Time taken to perform range query from zonemap = 1.44694e+07 microseconds
```

W5: W2 with unsorted data

```
[hot@scc-wi4 cs561_templatezonemaps2]$ ./workloadgenerator -N1000000 -R1000 -s0.001
[hot@scc-wi4 cs561_templatezonemaps2]$ ./main
Time taken to perform point queries from zonemap = 4944.67 microseconds
Time taken to perform range query from zonemap = 1.26579e+07 microseconds
```

W6: W3 with unsorted data

```
[hot@scc-wi4 cs561_templatezonemaps2]$ ./workloadgenerator -N1000000 -R1000 -s0.1
[hot@scc-wi4 cs561_templatezonemaps2]$ ./main
Time taken to perform point queries from zonemap = 4869.67 microseconds
Time taken to perform range query from zonemap = 1.15625e+07 microseconds
```

D)

1) What is the expected memory footprint (in bytes) to build the zone map? (assume: number of elements is N and every zone has d entries)
    a) Assume that the data being stored in the zonemaps are integers and that they are of size 4 bytes. In 1 zone we have the elements, min, max, and size, which should take 4*d bytes + 3* 4 bytes = 4*d + 12 bytes. In the zonemap, we have the total list of elements, total list of zones, number of zones, and number of elements per zone.
        i) Total List of elements = N * 4 bytes
        ii) Total List of Zones = (N/d) * (4*d + 12) bytes
            (1) Number of Zones = N/d
        iii) # of zones = 4 bytes
        iv) # of elements per zone = 4 bytes
        v) Memory footprint = (N*4) + (4N + 12N/d) + 8 bytes = 8N + 12N/d + 8 bytes
2) In practice, we may have raw data stored on disk, and the zone map is maintained in memory to reduce the number of required I/Os to answer queries. What should we do if we only have a limited memory budget to build the zone map (i.e., when the memory budget of M bytes is smaller than the expected memory footprint)?
    a) Based on the memory footprint we can attempt to increase the number of entries per zone, which would help shrink the amount of zones and subsequently metadata as well.
    b) Additionally there may be some zones that are queried more frequently than others, so we can potentially cache the more frequently queried zones in memory, and store less frequently used data within the disk to minimize the number of required I/Os.