Simulation for a priority queue with N events

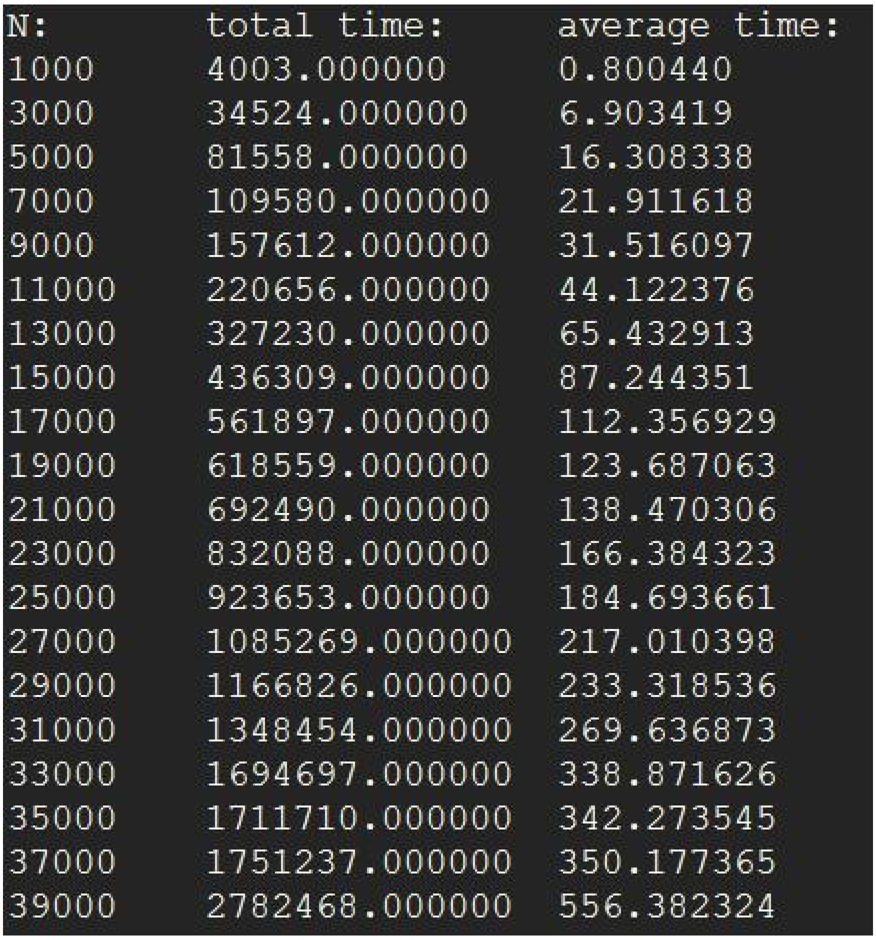
Chenwei Yue
GT ID 903243179

The definitions of most struct and functions are similar with that in the simulation program with three stations. But in this program, the events only have two attributes: a pointer to the next event and a timestamp. The timestamp of events are generated by the urand() function. After adding N events into the priority queue, the timer starts. The timer stops after 5000 loops of event insert and removal. I created two variables: struct timeval start and struct timeval finish. These variables have two important attributes: tv_usec and tv_sec, which records the microsecond and second of system time, respectively. So the operation time for this 5000 loops is:

duration=(finish.tv_sec -start.tv_sec) *1000000+(finish.tv_usec-start.tv_usec)

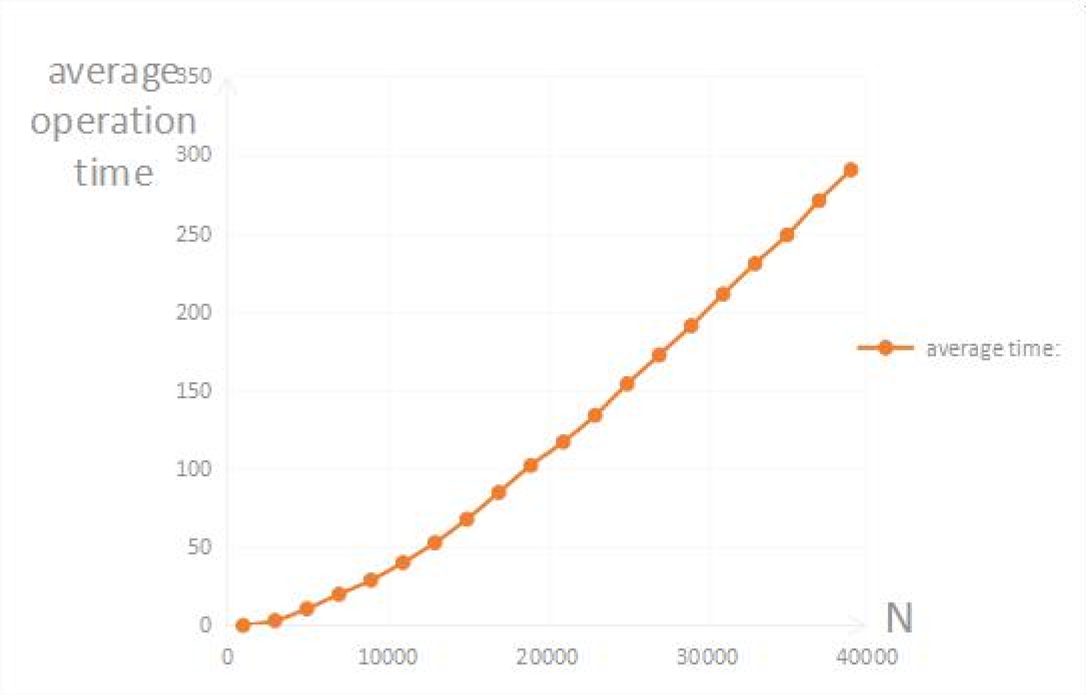
Here I used gettimeofday() function to record the system time.

When N increases, the average operation time for one iteration also increases. The results are shown below. (The unit of time is microsecond)



```
N:       total time:     average time:
1000     4003.000000     0.800440
3000     34524.000000    6.903419
5000     81558.000000    16.308338
7000     109580.000000   21.911618
9000     157612.000000   31.516097
11000    220656.000000   44.122376
13000    327230.000000   65.432913
15000    436309.000000   87.244351
17000    561897.000000   112.356929
19000    618559.000000   123.687063
21000    692490.000000   138.470306
23000    832088.000000   166.384323
25000    923653.000000   184.693661
27000    1085269.000000  217.010398
29000    1166826.000000  233.318536
31000    1348454.000000  269.636873
33000    1694697.000000  338.871626
35000    1711710.000000  342.273545
37000    1751237.000000  350.177365
39000    2782468.000000  556.382324
```

The graph for (N,average operation time) is:



According to this graph, we can conclude that the average operation time is approximately linearly related to N.