Name: Anh Pham
Student number: 050357871
Course: COMP.CS.300: Data Structures and Algorithms
Project 2

Table: Asymptotic performance of commands

Command	**Asymptotic performance and explanation**
clear_ways	O(n): Delete n ways
all_ways	O(n): Iterate through n ways
add_way *ID Coord1 Coord2 …*	O(n): Iterate through every coords to find total distance, then add to unordered_map(s)
ways_from *Coord*	O(n): Add n ways that have *Coord* as crossroad to a vector
way_coords *ID*	O(1): Find key in unorderd_map
route_any *Coord Coord*	O(V+E): BFS algorithm
remove_way *ID*	O(1): Find and erase element in 3 unordered_maps
route_least_crossroads *Coord Coord*	O(V+E): BFS algorithm
route_with cycle *Coord*	O(V+E): DFS algorithm
route_shortest_distance *Coord Coord*	O(V+ElogV): Dijkstra algorithm
trim_ways	O(n): Iterate through unordered_map

Struct for crossroads:

```
Struct Point {
     Coord
     Unordered_multimap crossroad: Store the crossroad's pair
     from WayID and distance of the way
     Distance d: Store the distance from the previous
     crossroad. Use in Dijkstra algorithm
     A pointer that points to the previous crossroad. Use when
     finding routes.
     A pointer that marks the node as unvisited, visited, or
     all its crossroads have been visited.
}
```

Data structures to store Ways and Crossroads

Unordered_map way_data: WayID as key, vector of cords as value

Unordered_map all_crossroads: Coord as key, pointer that point to that coord as value.

Reason for choosing data structures
None of the data needs to be stored in ordered, so unordered ones are enough. This help inserting, finding, and removing costs constant time on average.

Self-made functions:
Void reset_way_color(): Reset all crossroads to be unvisited after needed route is found and pointers to its correct value.
Distance way_distance(Way): Calculate a way’s distance of given vector of coords.
Distance total_way_distance: Calculate total distance of every ways. Intended to use in trim_ways command only

----

Testing output
Testing file output using command *testread “-in.txt” “-out.txt”* and other commands that I tested myself, compared to the expected output, there is no error.

Testing performance
Not many of the commands pass the perftest files, mainly due to adding time (in compulsory commands) or algorithm (find routes commands). A new file “perftest-all.txt” was created to perftest all commands for this phase

```
> read "perftest-all.txt"
** Commands from 'perftest-all.txt'
> perftest way_coords 20 5000
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 5000 random command(s) from:
way_coords

      N ,   add (sec) , cmds (sec) , total (sec)
     10 , 0.000287896 , 0.00812534 , 0.00841324
     30 , 0.000294062 , 0.00755763 , 0.00785169
    100 ,  0.00100497 , 0.00776657 , 0.00877154
    300 ,  0.00272821 , 0.00776433 ,  0.0104925
   1000 ,  0.00931108 ,  0.0079384 ,  0.0172495
   3000 ,   0.0291084 , 0.00751253 ,   0.036621
  10000 ,   0.0991203 , 0.00791541 ,   0.107036
  30000 ,    0.317263 , 0.00886548 ,   0.326128
 100000 ,     1.24175 ,  0.0100656 ,    1.25182
 300000 ,     4.10948 ,  0.0100055 ,    4.11948
```

```
1000000 ,     15.3147 ,  0.0117322 ,    15.3264
> perftest ways_from 20 5000
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 5000 random command(s) from:
ways_from

      N ,   add (sec) , cmds (sec) , total (sec)
     10 , 0.000128962 , 0.00551676 , 0.00564572
     30 , 0.000301497 , 0.00443259 , 0.00473409
    100 ,  0.00128116 , 0.00603886 , 0.00732002
    300 ,  0.00308531 , 0.00561757 , 0.00870288
   1000 ,   0.0113205 , 0.00670737 ,  0.0180278
   3000 ,   0.0371675 , 0.00788787 ,  0.0450554
  10000 ,    0.156701 , 0.00732422 ,   0.164025
  30000 ,    0.427905 , 0.00953678 ,   0.437442
 100000 ,     1.55653 ,  0.0111814 ,    1.56771
 300000 ,     4.57025 ,  0.0112163 ,    4.58147
1000000 ,     16.3956 ,  0.0161465 ,    16.4117
> perftest route_any 20 5000
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 5000 random command(s) from:
route_any

     N ,   add (sec) , cmds (sec) , total (sec)
    10 , 0.000141558 ,  0.0313842 ,   0.0315258
    30 , 0.000425526 ,   0.026232 ,   0.0266575
   100 ,  0.00143361 ,  0.0501033 ,   0.0515369
   300 ,  0.00320061 ,   0.114243 ,    0.117444
  1000 ,   0.0143946 ,   0.292042 ,    0.306437
  3000 ,   0.0335832 ,   0.878757 ,     0.91234
 10000 ,    0.125499 ,     3.1968 ,      3.3223
 30000 ,    0.462123 ,    15.5681 ,     16.0302
100000 ,     1.58171 , Timeout!
> perftest remove_way 20 5000
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 5000 random command(s) from:
remove_way

      N ,   add (sec) , cmds (sec) , total (sec)
     10 , 0.000163801 , 0.00835248 ,  0.00851629
     30 , 0.000361087 , 0.00578919 ,  0.00615028
    100 ,  0.00102715 , 0.00626265 ,   0.0072898
    300 ,  0.00345142 , 0.00798925 ,   0.0114407
   1000 ,   0.0151939 ,  0.0146213 ,   0.0298152
   3000 ,   0.0416611 ,  0.0222925 ,   0.0639536
  10000 ,    0.155198 ,  0.0318206 ,    0.187018
  30000 ,    0.430868 ,   0.032049 ,    0.462917
 100000 ,     1.59321 ,   0.039472 ,     1.63268
 300000 ,       5.097 ,  0.0366258 ,     5.13363
1000000 ,     17.5396 ,  0.0457132 ,     17.5853
> perftest route_least_crossroads 20 5000
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 5000 random command(s) from:
route_least_crossroads

     N ,   add (sec) , cmds (sec) , total (sec)
    10 , 0.000127943 ,  0.0347691 ,    0.034897
    30 , 0.000398379 ,  0.0239146 ,    0.024313
   100 ,   0.00113678,  0.0447237 ,   0.0458604
   300 ,  0.00456842 ,   0.129158 ,    0.133726
  1000 ,   0.0165028 ,   0.324947 ,     0.34145
  3000 ,   0.0340549 ,    1.12198 ,     1.15604
 10000 ,    0.133366 ,    4.05814 ,     4.19151
 30000 ,    0.418644 ,    17.0861 ,     17.5047
100000 ,     1.59964 , Timeout!
> perftest route_with_cycle 20 5000
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 5000 random command(s) from:
route_with_cycle

      N ,   add (sec) , cmds (sec) , total (sec)
     10 ,  0.00015547 ,  0.0498835 ,   0.0500389
     30 , 0.000517516 ,  0.0659304 ,   0.0664479
    100 ,  0.00113431 ,  0.0535872 ,   0.0547215
    300 ,  0.00346367 ,  0.0689381 ,   0.0724018
   1000 ,   0.0145317 ,  0.0909578 ,     0.10549
   3000 ,   0.0395883 ,   0.189028 ,    0.228617
  10000 ,    0.139815 ,   0.308349 ,    0.448164
  30000 ,    0.399191 ,   0.346534 ,    0.745725
 100000 ,     1.62473 ,   0.787343 ,     2.41207
 300000 ,     5.13129 ,    1.67039 ,     6.80168
1000000 ,     17.7805 , Timeout!
> perftest trim_ways 20 5000
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 5000 random command(s) from:
trim_ways

    N ,   add (sec) , cmds (sec) , total (sec)
   10 , 0.000162147 , 0.00814744 ,  0.00830958
   30 , 0.000534906 ,   0.044156 ,   0.0446909
  100 ,   0.0012107 ,  0.0452932 ,   0.0465039
  300 ,  0.00370625 ,   0.284409 ,    0.288115
 1000 ,   0.0158232 ,   0.952955 ,    0.968778
 3000 ,   0.0385143 ,    5.93949 ,       5.978
10000 ,    0.124187 , Timeout!
** End of commands from 'perftest-all.txt'
```