Course : TIE-20106
Student No.292119
Name : Nghia Duc Hong

Briefly asymptotic perfomance of self-made functions.

Function		Asymptotic complexity
stop_count()	1	Θ(1)
clear_all()	2	Θ(m) m is number of regions
all_stops()	3	Θ(n)
add_stop()	4	Ω(1) before*, then O(logn)
get_stop_name(id)	5	average O(1)
get_stop_coord(id)	6	average O(1)
stops_alphabetically()	7	O(nlogn) before*, then O(n)
stops_coord_order()	8	Most case O(nlogn), sometimes O(n)
min_coord()	9	Θ(1)
max_coord()	10	Θ(1)
find_stops(name)	11	O(nlogn) before*, then O(logn)
change_stop_name(id,newname)	12	average O(1) before *, then O(logn)
change_stop_coord(id,newcoord)	13	average Ω(1), O(n)
add_region(id, name)	14	average O(1)
get_region_name(id)	15	average O(1)
all_regions()	16	O(m): m is number of regions
add_stop_to_region(id,parentid)	17	average O(1)
add_subregion_to_region(id,p_id)	18	average O(1)
stop_regions(StopID id)	19	average O(m)
creation_finished()	20	Θ(1) : do nothing
region_bounding_box(RegionID id)	21	Θ(n)(n: N_points in that region)
stops_closest_to(StopID id)	22	Θ(n)
remove_stop(StopID id)	23	most case logn,remove max/min O(n)

stops_common_region(id1,id2)	24	Θ(min(h1,h2))

h is order of common region respect to id1,id2.

Space complexity : Θ(n)

Explain: event that the map that contains keys which are names is added, assume it is event*.
* only happens once;

The memory seems to work well, when I check memory leak by command line "valgrind --quiet --leak-check=full ./prg1.exe",
there is no memory problem, but when I check by qt-creator, it shows that the program has serveral problems. When I
check the program with only command quit, it has 29 problems, and when I run the plain program with out my data structures
with only quit command, it shows there is 28 problems, so I think those problem belong to the other files of the program.

Describe datastructes :
Struct : Point
{
StopID
Name
Coord
RegionID
V_ptr : the pointer that the vector contains coresponding to point.
}
Region
{
RegionID
Name
unordered_map subpoints;
unordered_map subregions;
parent_region;
}
A unordered_map has keys are the StopID, and the values are the pointer point to the struct Stop.(mp)
A ordered_map has keys are Stop Name, the values are id. (namemap)
A vector has pointers to the pair<StopID, Coord>. (id_to_coordinate)
A unordered_map has keys are regionsID, and the values are pointer to region. (

The unoreder_map mp makes access to StopID with average constant time
The map namemap makes reduce complexity of some functions and increase complexity of some functions.
But overal the time is balanced between functions.
The vector id_to_coordinate : the vector always is ready to be sorted, by managing it properly, so
accessing, removing and searching element in vector is average constant time.

```
TESTING
--------------------------------------------------------------------------------
--------------------------------------------------------
> testread "example-compulsory-in.txt" "example-compulsory-out.txt"
  Actual output                                  | Expected output
-------------------------------------------------|------------------------
---------------------
  > clear_all                                    | > clear_all
  Cleared everything.                            | Cleared everything.
  > stop_count                                   | > stop_count
  Number of stops: 0                             | Number of stops: 0
  > read "example-stops.txt"                     | > read "example-
stops.txt"
  ** Commands from 'example-stops.txt'           | ** Commands from
'example-stops.txt'
  > # Add stops                                  | > # Add stops
  > add_stop 1 One (1,1)                         | > add_stop 1 One (1,1)
  One: pos=(1,1), id=1                           | One: pos=(1,1), id=1
  > add_stop 2 Two (6,2)                         | > add_stop 2 Two (6,2)
  Two: pos=(6,2), id=2                           | Two: pos=(6,2), id=2
  > add_stop 3 Three (0,6)                       | > add_stop 3 Three (0,6)
  Three: pos=(0,6), id=3                         | Three: pos=(0,6), id=3
  > add_stop 4 Four (7,7)                        | > add_stop 4 Four (7,7)
  Four: pos=(7,7), id=4                          | Four: pos=(7,7), id=4
  > add_stop 5 Five (4,4)                        | > add_stop 5 Five (4,4)
  Five: pos=(4,4), id=5                          | Five: pos=(4,4), id=5
  > add_stop 6 Six (2,9)                         | > add_stop 6 Six (2,9)
  Six: pos=(2,9), id=6                           | Six: pos=(2,9), id=6
  >                                              | >
  ** End of commands from 'example-stops.txt'    | ** End of commands from
'example-stops.txt'
  > stop_count                                   | > stop_count
  Number of stops: 6                             | Number of stops: 6
  > stop_name 1                                  | > stop_name 1
  Stop ID 1 has name 'One'                       | Stop ID 1 has name 'One'
  One: pos=(1,1), id=1                           | One: pos=(1,1), id=1
  > stop_coord 5                                 | > stop_coord 5
  Stop ID 5 is in position (4,4)                 | Stop ID 5 is in position
(4,4)
  Five: pos=(4,4), id=5                          | Five: pos=(4,4), id=5
  > stops_alphabetically                         | > stops_alphabetically
  1.  Five: pos=(4,4), id=5                       | 1. Five: pos=(4,4),
      id=5
  2.  Four: pos=(7,7), id=4                       | 2. Four: pos=(7,7),
      id=4
  3.  One: pos=(1,1), id=1                        | 3. One: pos=(1,1), id=1
  4.  Six: pos=(2,9), id=6                        | 4. Six: pos=(2,9), id=6
  5.  Three: pos=(0,6), id=3                      | 5. Three: pos=(0,6),
      id=3
  6.  Two: pos=(6,2), id=2                        | 6. Two: pos=(6,2), id=2
  > min_coord                                    | > min_coord
  One: pos=(1,1), id=1                           | One: pos=(1,1), id=1
  > max_coord                                    | > max_coord
```

```
  Four: pos=(7,7), id=4                          | Four: pos=(7,7), id=4
  > stops_coord_order                            | > stops_coord_order
  1.  One: pos=(1,1), id=1                        | 1. One: pos=(1,1), id=1
  2.  Five: pos=(4,4), id=5                       | 2. Five: pos=(4,4),
      id=5
  3.  Three: pos=(0,6), id=3                      | 3. Three: pos=(0,6),
      id=3
  4.  Two: pos=(6,2), id=2                        | 4. Two: pos=(6,2), id=2
  5.  Six: pos=(2,9), id=6                        | 5. Six: pos=(2,9), id=6
  6.  Four: pos=(7,7), id=4                       | 6. Four: pos=(7,7),
      id=4
  > change_stop_name 5 Two                       | > change_stop_name 5 Two
  Two: pos=(4,4), id=5                           | Two: pos=(4,4), id=5
  > find_stops Two                               | > find_stops Two
  1.  Two: pos=(6,2), id=2                        | 1. Two: pos=(6,2), id=2
  2.  Two: pos=(4,4), id=5                        | 2. Two: pos=(4,4), id=5
  > read "example-regions.txt"                   | > read "example-
regions.txt"
  ** Commands from 'example-regions.txt'         | ** Commands from
'example-regions.txt'
  > # Add regions and stops to regions           | > # Add regions and
stops to regions
  > add_region S Small                           | > add_region S Small
  Region: Small: id=S                            | Region: Small: id=S
  > add_stop_to_region 1 S                       | > add_stop_to_region 1 S
  Added stop One to region Small                 | Added stop One to region
Small
  Region: Small: id=S                            | Region: Small: id=S
  One: pos=(1,1), id=1                           | One: pos=(1,1), id=1
  > add_stop_to_region 2 S                       | > add_stop_to_region 2 S
  Added stop Two to region Small                 | Added stop Two to region
Small
  Region: Small: id=S                            | Region: Small: id=S
  Two: pos=(6,2), id=2                           | Two: pos=(6,2), id=2
  > add_stop_to_region 3 S                       | > add_stop_to_region 3 S
  Added stop Three to region Small               | Added stop Three to
region Small
  Region: Small: id=S                            | Region: Small: id=S
  Three: pos=(0,6), id=3                         | Three: pos=(0,6), id=3
  > add_region L Large                           | > add_region L Large
  Region: Large: id=L                            | Region: Large: id=L
  > add_subregion_to_region S L                  | >
add_subregion_to_region S L
  Added subregion Small to region Large          | Added subregion Small to
region Large
  > add_stop_to_region 4 L                       | > add_stop_to_region 4 L
  Added stop Four to region Large                | Added stop Four to
region Large
  Region: Large: id=L                            | Region: Large: id=L
  Four: pos=(7,7), id=4                          | Four: pos=(7,7), id=4
  > add_stop_to_region 5 L                       | > add_stop_to_region 5 L
  Added stop Two to region Large                 | Added stop Two to region
Large
  Region: Large: id=L                            | Region: Large: id=L
```

```
  Two: pos=(4,4), id=5                              | Two: pos=(4,4), id=5
  > add_stop_to_region 6 L                          | > add_stop_to_region 6 L
  Added stop Six to region Large                    | Added stop Six to region
Large
  Region: Large: id=L                               | Region: Large: id=L
  Six: pos=(2,9), id=6                              | Six: pos=(2,9), id=6
  >                                                 | >
  ** End of commands from 'example-regions.txt'     | ** End of commands from
'example-regions.txt'
  > all_regions                                     | > all_regions
  1.  Large: id=L                                    | 1. Large: id=L
  2.  Small: id=S                                    | 2. Small: id=S
  > region_name S                                   | > region_name S
  Region ID S has name 'Small'                      | Region ID S has name
'Small'
  Small: id=S                                       | Small: id=S
  > stop_regions 1                                  | > stop_regions 1
  Regions for stop One: pos=(1,1), id=1             | Regions for stop One:
pos=(1,1), id=1
  1.  Small: id=S                                    | 1. Small: id=S
  2.  Large: id=L                                    | 2. Large: id=L
  > quit                                            | > quit
**No differences in output.**
Testread-tests have been run, no differences found.
--------------------------------------------------------------------------------
-------------------------------------------
> read "perftest-all.txt"
** Commands from 'perftest-all.txt'
> # Read performance tests of all operations
> read "perftest-compulsory.txt"
** Commands from 'perftest-compulsory.txt'
> # Read performance tests of compulsory operations
> read "perftest-access.txt"
** Commands from 'perftest-access.txt'
> # Test the performance of stop_name/stop_coord/region_name
> perftest stop_name;stop_coord;region_name 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:
stop_name stop_coord region_name

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   7.5146e-05 ,   0.00352438 ,   0.00359952
     30 ,  0.000169947 ,   0.00343033 ,   0.00360028
    100 ,  0.000604873 ,   0.00340884 ,   0.00401371
    300 ,    0.0017469 ,    0.0035872 ,   0.00533411
   1000 ,   0.00608844 ,   0.00366964 ,   0.00975809
   3000 ,    0.0196319 ,   0.00430459 ,    0.0239365
  10000 ,    0.0731991 ,   0.00486996 ,     0.078069
  30000 ,      0.25662 ,   0.00511146 ,     0.261732
 100000 ,     0.924523 ,   0.00618124 ,     0.930704
 300000 ,      3.19691 ,   0.00617603 ,      3.20309
1000000 ,      10.9585 ,   0.00628322 ,      10.9648
>
```

```
** End of commands from 'perftest-access.txt'
> read "perftest-sorting.txt"
** Commands from 'perftest-sorting.txt'
> # Test the performance of sorting, adding stops in between
> perftest stops_alphabetically;stops_coord_order;random_add 20 500
10;30;100;300;1000;3000;10000;30000;100000;300000
Timeout for each N is 20 sec.
For each N perform 500 random command(s) from:
stops_alphabetically stops_coord_order random_add

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   8.4995e-05 ,   0.00753877 ,   0.00762377
     30 ,   0.00018824 ,   0.00943628 ,   0.00962452
    100 ,  0.000577426 ,     0.017902 ,    0.0184794
    300 ,   0.00162601 ,    0.0459415 ,    0.0475675
   1000 ,   0.00565528 ,     0.125615 ,      0.13127
   3000 ,    0.0197191 ,      0.63446 ,     0.654179
  10000 ,     0.077474 ,      2.31979 ,      2.39726
  30000 ,     0.304167 ,      9.92345 ,      10.2276
 100000 ,     0.807736 , Timeout!
>
** End of commands from 'perftest-sorting.txt'
> read "perftest-minmax.txt"
** Commands from 'perftest-minmax.txt'
> # Test the performance of min/max, adding stops in between
> perftest min_coord;max_coord;random_add 20 500
10;30;100;300;1000;3000;10000;30000;100000;300000;1000000
Timeout for each N is 20 sec.
For each N perform 500 random command(s) from:
min_coord max_coord random_add

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   8.2211e-05 ,   0.00118637 ,   0.00126858
     30 ,  0.000248777 ,   0.00118909 ,   0.00143787
    100 ,  0.000577271 ,    0.0012149 ,   0.00179217
    300 ,   0.00164373 ,   0.00967544 ,    0.0113192
   1000 ,    0.0208277 ,   0.00115683 ,    0.0219845
   3000 ,    0.0266358 ,   0.00138869 ,    0.0280245
  10000 ,    0.0922748 ,   0.00146542 ,    0.0937402
  30000 ,     0.231159 ,   0.00169711 ,     0.232856
 100000 ,     0.834754 ,   0.00168054 ,     0.836434
 300000 ,      2.68672 ,   0.00178436 ,      2.68851
1000000 ,      9.88545 ,   0.00201068 ,      9.88746
>
** End of commands from 'perftest-minmax.txt'
> read "perftest-change.txt"
** Commands from 'perftest-change.txt'
> # Test the performance of change_stop_name/change_stop_coord
> perftest change_stop_name;change_stop_coord 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:
change_stop_name change_stop_coord
```

```
      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   8.6678e-05 ,   0.00271401 ,   0.00280069
     30 ,  0.000192402 ,   0.00265504 ,   0.00284744
    100 ,  0.000586946 ,   0.00267234 ,   0.00325929
    300 ,   0.00162167 ,   0.00270345 ,   0.00432512
   1000 ,   0.00593193 ,   0.00273853 ,   0.00867046
   3000 ,    0.0188344 ,   0.00278584 ,    0.0216202
  10000 ,    0.0885217 ,   0.00294707 ,    0.0914687
  30000 ,     0.238793 ,    0.0032587 ,     0.242052
 100000 ,     0.822733 ,   0.00353278 ,     0.826266
 300000 ,      2.71001 ,   0.00408615 ,       2.7141
1000000 ,      10.8668 ,   0.00575608 ,      10.8726
>
** End of commands from 'perftest-change.txt'
> read "perftest-stop_regions.txt"
** Commands from 'perftest-stop_regions.txt'
> # Test the performance of stop_regions
> perftest stop_regions 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:
stop_regions

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   9.4577e-05 ,   0.00633478 ,   0.00642935
     30 ,  0.000205484 ,   0.00837477 ,   0.00858026
    100 ,  0.000649077 ,    0.0104863 ,    0.0111353
    300 ,   0.00178139 ,    0.0129121 ,    0.0146934
   1000 ,   0.00564908 ,    0.0160853 ,    0.0217344
   3000 ,    0.0198908 ,    0.0226815 ,    0.0425722
  10000 ,    0.0700711 ,    0.0281303 ,    0.0982014
  30000 ,     0.252531 ,    0.0355299 ,     0.288061
 100000 ,     0.920922 ,    0.0407023 ,     0.961625
 300000 ,      3.07657 ,     0.043839 ,      3.12041
1000000 ,      10.9496 ,    0.0474393 ,      10.9971
>
** End of commands from 'perftest-stop_regions.txt'
> read "perftest-find_stops.txt"
** Commands from 'perftest-find_stops.txt'
> # Test the performance of find_stops
> perftest find_stops 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:
find_stops

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   9.4648e-05 ,    0.0073549 ,   0.00744955
     30 ,  0.000206666 ,   0.00803099 ,   0.00823765
    100 ,  0.000611559 ,   0.00905133 ,   0.00966288
    300 ,   0.00178095 ,    0.0108053 ,    0.0125863
   1000 ,   0.00580739 ,    0.0113958 ,    0.0172032
   3000 ,    0.0185027 ,    0.0131848 ,    0.0316875
  10000 ,    0.0747642 ,    0.0161976 ,    0.0909618
```

```
  30000 ,     0.241515 ,     0.019056 ,     0.260571
 100000 ,     0.895109 ,    0.0222701 ,     0.917379
 300000 ,      3.08057 ,    0.0288839 ,      3.10945
1000000 ,      11.2956 ,    0.0305745 ,      11.3261
>
** End of commands from 'perftest-find_stops.txt'
>
** End of commands from 'perftest-compulsory.txt'
> read "perftest-bbox.txt"
** Commands from 'perftest-bbox.txt'
> # Test the performance of region_bounding_box
> perftest region_bounding_box 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:
region_bounding_box

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   0.00016202 ,   0.00946754 ,   0.00962956
     30 ,  0.000193835 ,     0.012996 ,    0.0131898
    100 ,  0.000600257 ,    0.0164369 ,    0.0170372
    300 ,   0.00170556 ,    0.0233309 ,    0.0250364
   1000 ,   0.00576052 ,    0.0301656 ,    0.0359261
   3000 ,    0.0177585 ,     0.042103 ,    0.0598615
  10000 ,    0.0648045 ,    0.0499951 ,       0.1148
  30000 ,     0.224489 ,    0.0851372 ,     0.309626
 100000 ,     0.935403 ,     0.315731 ,      1.25113
 300000 ,      3.20362 ,     0.213002 ,      3.41662
1000000 ,      11.0171 ,     0.265545 ,      11.2827
>
** End of commands from 'perftest-bbox.txt'
> read "perftest-stops_closest_to.txt"
** Commands from 'perftest-stops_closest_to.txt'
> # Test the performance of stops_closest_to
> perftest stops_closest_to 20 500
10;30;100;300;1000;3000;10000;30000;100000;300000
Timeout for each N is 20 sec.
For each N perform 500 random command(s) from:
stops_closest_to

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,  0.000149888 ,  0.000777888 ,  0.000927776
     30 ,  0.000188927 ,  0.000899207 ,   0.00108813
    100 ,   0.00071546 ,  0.000993819 ,   0.00170928
    300 ,   0.00175269 ,   0.00153876 ,   0.00329145
   1000 ,   0.00576556 ,   0.00186488 ,   0.00763044
   3000 ,    0.0184499 ,   0.00226512 ,     0.020715
  10000 ,    0.0669749 ,   0.00273271 ,    0.0697076
  30000 ,     0.220878 ,   0.00321248 ,      0.22409
 100000 ,     0.896127 ,    0.0039482 ,     0.900075
 300000 ,      2.87159 ,   0.00429948 ,      2.87589
>
** End of commands from 'perftest-stops_closest_to.txt'
> read "perftest-stops_common_region.txt"
```

```
** Commands from 'perftest-stops_common_region.txt'
> # Test the performance of stops_common_region
> perftest stops_common_region 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:
stops_common_region

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,  0.000173384 ,    0.0151624 ,    0.0153358
     30 ,  0.000179852 ,    0.0181065 ,    0.0182863
    100 ,  0.000569441 ,    0.0244947 ,    0.0250642
    300 ,   0.00170581 ,    0.0310248 ,    0.0327306
   1000 ,   0.00592742 ,    0.0413992 ,    0.0473266
   3000 ,    0.0180707 ,    0.0493111 ,    0.0673818
  10000 ,    0.0646823 ,    0.0608083 ,     0.125491
  30000 ,     0.211588 ,    0.0719511 ,     0.283539
 100000 ,     0.851093 ,    0.0909385 ,     0.942031
 300000 ,      2.90088 ,     0.101891 ,      3.00277
1000000 ,      10.5352 ,     0.122645 ,      10.6579
>
** End of commands from 'perftest-stops_common_region.txt'
> read "perftest-remove.txt"
** Commands from 'perftest-remove.txt'
> # Test the performance of remove_stop
> perftest remove_stop 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_stop

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   9.7962e-05 ,   0.00369768 ,   0.00379564
     30 ,  0.000213287 ,   0.00320606 ,   0.00341934
    100 ,  0.000694168 ,   0.00391344 ,   0.00460761
    300 ,   0.00175309 ,   0.00430325 ,   0.00605635
   1000 ,   0.00577287 ,   0.00552178 ,    0.0112946
   3000 ,    0.0179958 ,    0.0294283 ,    0.0474241
  10000 ,    0.0692118 ,    0.0269408 ,    0.0961526
  30000 ,     0.241033 ,    0.0371299 ,     0.278163
 100000 ,     0.883316 ,    0.0404524 ,     0.923768
 300000 ,      2.90919 ,    0.0522359 ,      2.96143
1000000 ,      10.5182 ,    0.0515103 ,      10.5698
>
** End of commands from 'perftest-remove.txt'
>
** End of commands from 'perftest-all.txt'
```