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) if * hasnot happens, average O(1)

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

In testread "Treetest-all-in.txt" "Tretest-all-out.txt", there is only different in listing stops alphabetically because they have the same name.

```
------------------------------------------------------------------------
------------------------------------------------------------
> 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 ,    8.452e-05 ,   0.00326234 ,   0.00334686
     30 ,   0.00014462 ,   0.00330112 ,   0.00344574
    100 ,  0.000411974 ,   0.00323639 ,   0.00364837
    300 ,   0.00123581 ,   0.00342089 ,    0.0046567
   1000 ,   0.00431014 ,   0.00340177 ,   0.00771191
   3000 ,    0.0131303 ,    0.0036895 ,    0.0168198
  10000 ,    0.0466197 ,   0.00409144 ,    0.0507111
  30000 ,     0.155693 ,   0.00427735 ,     0.159971
 100000 ,     0.532941 ,   0.00513626 ,     0.538077
 300000 ,      1.74508 ,   0.00525229 ,      1.75034
1000000 ,      6.13245 ,   0.00539171 ,      6.13784
>
** 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 ,   5.8039e-05 ,   0.00875054 ,   0.00880858
     30 ,  0.000140175 ,   0.00916462 ,   0.00930479
    100 ,  0.000400705 ,    0.0181212 ,    0.0185219
    300 ,   0.00122292 ,    0.0420468 ,    0.0432697
```

```
   1000 ,   0.00412268 ,     0.121951 ,     0.126074
   3000 ,    0.0126547 ,     0.422777 ,     0.435432
  10000 ,      0.04377 ,      1.60529 ,      1.64906
  30000 ,     0.138461 ,      6.22554 ,        6.364
 100000 ,     0.494881 , 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 ,   5.8032e-05 ,  0.000811961 ,  0.000869993
     30 ,  0.000123395 ,  0.000825707 ,  0.000949102
    100 ,  0.000395752 ,  0.000804025 ,   0.00119978
    300 ,   0.00120787 ,  0.000862338 ,   0.00207021
   1000 ,   0.00406426 ,  0.000865258 ,   0.00492952
   3000 ,      0.01255 ,  0.000822278 ,    0.0133723
  10000 ,    0.0438583 ,  0.000938326 ,    0.0447966
  30000 ,     0.135835 ,  0.000876472 ,     0.136712
 100000 ,     0.497989 ,  0.000950088 ,     0.498939
 300000 ,      1.53483 ,  0.000865179 ,      1.53569
1000000 ,      5.50286 ,   0.00120112 ,      5.50406
>
** 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 ,  0.000128388 ,   0.00423927 ,   0.00436766
     30 ,  0.000151343 ,   0.00253913 ,   0.00269047
    100 ,  0.000417332 ,   0.00251765 ,   0.00293498
    300 ,   0.00125075 ,   0.00259739 ,   0.00384814
   1000 ,    0.0041023 ,   0.00264313 ,   0.00674543
   3000 ,    0.0125499 ,   0.00267509 ,     0.015225
  10000 ,    0.0433778 ,   0.00279507 ,    0.0461729
  30000 ,     0.134562 ,   0.00301353 ,     0.137575
 100000 ,     0.489138 ,   0.00333711 ,     0.492475
 300000 ,      1.54555 ,    0.0036469 ,       1.5492
1000000 ,      5.49394 ,   0.00462589 ,      5.49857
>
** 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 ,   5.6837e-05 ,   0.00595905 ,   0.00601589
     30 ,  0.000136122 ,   0.00721431 ,   0.00735044
    100 ,  0.000441286 ,   0.00904637 ,   0.00948766
    300 ,   0.00125113 ,       0.0127 ,    0.0139511
   1000 ,    0.0041322 ,    0.0138896 ,    0.0180218
   3000 ,    0.0126129 ,     0.018789 ,     0.031402
  10000 ,    0.0432411 ,    0.0232778 ,    0.0665189
  30000 ,     0.136933 ,    0.0307205 ,     0.167654
 100000 ,     0.496111 ,    0.0318011 ,     0.527912
 300000 ,      1.56606 ,    0.0378621 ,      1.60393
1000000 ,      5.51456 ,    0.0428816 ,      5.55745
>
** 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 ,   5.5597e-05 ,   0.00684552 ,   0.00690112
     30 ,  0.000136694 ,   0.00770859 ,   0.00784528
    100 ,  0.000437487 ,   0.00866602 ,    0.0091035
    300 ,   0.00124509 ,   0.00979539 ,    0.0110405
   1000 ,   0.00416521 ,    0.0110405 ,    0.0152058
   3000 ,    0.0125771 ,    0.0143665 ,    0.0269436
  10000 ,    0.0439827 ,    0.0247578 ,    0.0687405
  30000 ,     0.140387 ,    0.0539533 ,     0.194341
 100000 ,       0.5301 ,     0.163364 ,     0.693465
 300000 ,      1.56542 ,     0.545549 ,      2.11097
1000000 ,      5.48114 ,      1.80976 ,      7.29089
>
** 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 ,   6.4774e-05 ,   0.00896101 ,   0.00902579
     30 ,  0.000145648 ,    0.0117691 ,    0.0119147
    100 ,  0.000442661 ,    0.0154508 ,    0.0158934
    300 ,   0.00129697 ,    0.0195155 ,    0.0208125
   1000 ,   0.00417736 ,    0.0282531 ,    0.0324305
   3000 ,    0.0128023 ,    0.0308858 ,    0.0436881
  10000 ,    0.0435462 ,    0.0519652 ,    0.0955114
  30000 ,     0.138237 ,    0.0945995 ,     0.232836
 100000 ,     0.494723 ,     0.197635 ,     0.692358
 300000 ,      1.57042 ,     0.198076 ,      1.76849
1000000 ,      5.47886 ,     0.163002 ,      5.64186
>
** 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 ,   5.7871e-05 ,  0.000641797 ,  0.000699668
     30 ,  0.000134931 ,  0.000737076 ,  0.000872007
    100 ,  0.000407703 ,   0.00101461 ,   0.00142231
    300 ,   0.00123175 ,   0.00120798 ,   0.00243973
   1000 ,   0.00407104 ,   0.00173118 ,   0.00580223
   3000 ,    0.0124384 ,   0.00196413 ,    0.0144025
  10000 ,    0.0432958 ,   0.00253597 ,    0.0458317
  30000 ,     0.136269 ,   0.00280316 ,     0.139072
 100000 ,     0.489961 ,   0.00351237 ,     0.493473
 300000 ,      1.55021 ,   0.00385657 ,      1.55407
>
** 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 ,   6.4908e-05 ,    0.0141438 ,    0.0142087
     30 ,  0.000137848 ,     0.017229 ,    0.0173668
    100 ,  0.000433787 ,    0.0231623 ,    0.0235961
    300 ,   0.00127762 ,    0.0290314 ,     0.030309
   1000 ,   0.00416058 ,    0.0368388 ,    0.0409994
   3000 ,    0.0125339 ,    0.0438854 ,    0.0564194
```

```
  10000 ,    0.0434066 ,    0.0586986 ,     0.102105
  30000 ,     0.135597 ,    0.0707154 ,     0.206313
 100000 ,     0.489157 ,    0.0766476 ,     0.565805
 300000 ,      1.55776 ,    0.0921189 ,      1.64988
1000000 ,      5.68906 ,     0.112898 ,      5.80196
>
** 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 ,   5.5796e-05 ,   0.00352787 ,   0.00358366
     30 ,  0.000147618 ,   0.00334538 ,   0.00349299
    100 ,  0.000440711 ,   0.00343557 ,   0.00387629
    300 ,   0.00128941 ,   0.00400944 ,   0.00529885
   1000 ,   0.00455355 ,   0.00362852 ,   0.00818207
   3000 ,    0.0126554 ,   0.00675378 ,    0.0194092
  10000 ,    0.0458963 ,    0.0125775 ,    0.0584738
  30000 ,     0.135626 ,    0.0150877 ,     0.150714
 100000 ,     0.491222 ,    0.0173114 ,     0.508533
 300000 ,      1.55488 ,    0.0250263 ,      1.57991
1000000 ,      5.51907 ,    0.0207357 ,      5.53981
>
** End of commands from 'perftest-remove.txt'
>
** End of commands from 'perftest-all.txt'
```