Name: Anh Pham
Student number: 050357871
Course: COMP.CS.300: Data Structures and Algorithms
Project 1

Table: Asymptotic performance of commands

Command	**Asymptotic performance and explanation**
place_count	Θ(1): Calculate vector's size
clear_all	Θ(n): n is number of places/areas
all_places	Θ(n): iterate through all elements of vector
add_place *ID 'Name' type (x,y)*	O(logn): Add new element to 2 multimaps and 1 unordered map
place_name_type *ID*	Θ(1): Search unordered_map with key
place_coord *ID*	Θ(1): Search unordered_map with key
places_alphabetically	Θ(n): Iterate through a multimap
places_coord_order	Θ(nlogn): Sort a vector using std::sort
find_places_name *name*	O(logn) then log n: Find positions using equal_range(), and add to a vector
find_places_type *type*	O(n): Search an unordered_map O(log7),then copy elements from an unordered_set to a vector
change_place_name *ID newname*	O(logn): Search, erase, add element in map
change_place_coord *ID (x,y)*	O(1): Search and modify value in unordered_map
add_area *ID Name Coord1 Coord2*	O(1): Add new to unordered_map
area_name *ID*	O(1): Search unordered_map with key
area_coords *ID*	O(1): Search unordered_map with key
all_areas	Θ(n): Iterate through unordered_map
add_subarea_to_area *AreaID AreaID*	O(1): Search and assign in unordered_map
subarea_in_areas *AreaID*	O(k) (k: number of parent areas) Continuously looking for parent area using pointer
creation_finished	Θ(1): Do nothing

all_Subareas_in_area *AreaID*	O(m^k) (m: number of direct subareas; the last subarea of areaid is the kth area): Area pointer continuously search if its parent area exists, and recursively with other direct subareas
places_closest_to *Coord [type]*	O(n) then O(m) (m: number of places have the same type): partition a vector and sort a constant amount
remove_place *ID*	O(logn): Delete key in unordered_map: O(1), value in multimap: O(logn), element in unordered_set: O(1)
common_area_of_subareas *AreaID AreaID*	Θ(min(n1,n2)) (n: number of common parent areas): search for the first same pointer

---

Data structures for place and area:

```
struct Place{
     PlaceID
     Name
     PlaceType
     Coord
     Id_TypeCoord_ptr: Pointer that points to a place data
     (with id, type and coord)
}

struct Area{
     AreaID
     Name
     vector areaboundary: contains all coordinates of an area
     shared_ptr parent area: point to its parent area
     unordered_map subareas: each element in map is a subarea,
     with AreaID as key and a pointer points to Area as value
}
```

Data structures to store places and areas' data:

multimap placename_to_placeid: name as key, id as value
unordered_map placetype_to_placeid: type as key,
unordered_set of ids as value
unordered_map placeid_to_data: id as key, pointer that points to place as value
vector placeid_to_placetypecoord: contains Id_TypeCoord_ptr mentioned above

unordered_map areaid_to_data: id as key, pointer that points to an area as value

Reasons for choosing above data structures:

unordered_map(s) for place and area let me access Place or Area with constant time.

multimap placename_to_place_id increases performance on find_places_name command. This affects add_place performance a little, but it is balanced between commands.

unordered_map placetype_to_place_id allows command find_place_type cost O(n) for copying, and remove_place costs O(1)

vector placeid_to_placetypecoord is always sorted for better adding, searching, and removing performance.

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

Some commands have better performance while others have worse than expected. Timeout when testing remove_place and sorting commands.

```
> 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 place_name_type;place_coord;area_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:
place_name_type place_coord area_name

      N ,   add (sec) , cmds (sec) , total (sec)
     10 ,   8.754e-05 , 0.00318109 , 0.00326863
     30 , 0.000219598 , 0.00337709 , 0.00359669
    100 , 0.000594732 , 0.00230674 , 0.00290147
    300 ,  0.00102556 , 0.00225615 , 0.00328172
   1000 ,  0.00370156 , 0.00261628 , 0.00631784
   3000 ,   0.0122888 , 0.00283437 ,  0.0151232
  10000 ,     0.03975 , 0.00280426 ,  0.0425543
  30000 ,    0.130457 , 0.00396234 ,   0.134419
```

```
 100000 ,    0.518952 , 0.00401682 ,   0.522969
 300000 ,     1.63807 , 0.00463138 ,     1.6427
1000000 ,     6.49788 , 0.00539017 ,    6.50327
>
** 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 places_alphabetically;places_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:
places_alphabetically places_coord_order random_add

     N ,   add (sec) , cmds (sec) , total (sec)
    10 ,  5.5324e-05 ,  0.0032598 , 0.00331512
    30 , 0.000160722 , 0.00433405 , 0.00449477
   100 , 0.000520889 , 0.00631828 , 0.00683917
   300 ,  0.00138481 ,  0.0129996 ,  0.0143844
  1000 ,  0.00434403 ,  0.0350171 ,  0.0393612
  3000 ,   0.0147307 ,   0.130844 ,   0.145574
 10000 ,   0.0651176 ,   0.398661 ,   0.463778
 30000 ,    0.167235 ,    2.09773 ,    2.26497
100000 ,    0.773338 ,    8.24666 ,       9.02
300000 ,     2.20213 , Timeout!
>
** End of commands from 'perftest-sorting.txt'
> read "perftest-change.txt"
** Commands from 'perftest-change.txt'
> # Test the performance of changing name/coord
> perftest change_place_name;change_place_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_place_name change_place_coord

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,  0.000138783 ,     0.013141 ,    0.0132797
     30 ,  0.000290715 ,    0.0143579 ,    0.0146487
    100 ,  0.000800245 ,    0.0169585 ,    0.0177587
    300 ,   0.00232342 ,    0.0195148 ,    0.0218382
   1000 ,   0.00743485 ,    0.0213982 ,    0.0288331
   3000 ,    0.0236937 ,    0.0253586 ,    0.0490523
  10000 ,    0.0502909 ,    0.0159476 ,    0.0662385
  30000 ,     0.265598 ,    0.0234876 ,     0.289086
 100000 ,     0.811605 ,    0.0247013 ,     0.836307
 300000 ,      1.87684 ,    0.0229995 ,      1.89984
1000000 ,      6.40891 ,    0.0272518 ,      6.43616
>
** End of commands from 'perftest-change.txt'
> read "perftest-find_places.txt"
```

```
** Commands from 'perftest-find_places.txt'
> # Test the performance of finding places
> perftest find_places_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:
find_places_name

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   9.0791e-05 ,   0.00493246 ,   0.00502325
     30 ,  0.000154556 ,   0.00669964 ,   0.00685419
    100 ,  0.000415984 ,   0.00730754 ,   0.00772353
    300 ,   0.00115627 ,   0.00754763 ,    0.0087039
   1000 ,   0.00389079 ,    0.0093723 ,    0.0132631
   3000 ,    0.0126549 ,    0.0093344 ,    0.0219892
  10000 ,    0.0431546 ,    0.0115775 ,     0.054732
  30000 ,     0.224112 ,    0.0290181 ,      0.25313
 100000 ,      0.62064 ,    0.0175487 ,     0.638188
 300000 ,      1.80719 ,    0.0217142 ,      1.82891
1000000 ,      6.47738 ,     0.036502 ,      6.51388
> perftest find_places_type 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_places_type

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,  0.000107426 ,   0.00331276 ,   0.00342019
     30 ,  0.000176868 ,   0.00371062 ,   0.00388749
    100 ,  0.000517351 ,   0.00436254 ,   0.00487989
    300 ,   0.00139458 ,   0.00601666 ,   0.00741124
   1000 ,   0.00474985 ,   0.00546885 ,    0.0102187
   3000 ,      0.01453 ,   0.00476553 ,    0.0192955
  10000 ,    0.0610432 ,    0.0122556 ,    0.0732988
  30000 ,      0.23571 ,    0.0155349 ,     0.251245
 100000 ,     0.607223 ,     0.054121 ,     0.661344
 300000 ,      1.85898 ,     0.223524 ,      2.08251
1000000 ,      6.29888 ,     0.662845 ,      6.96172
>
** End of commands from 'perftest-find_places.txt'
> read "perftest-subareas.txt"
** Commands from 'perftest-subareas.txt'
> # Test the performance of subarea chain
> perftest subarea_in_areas 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:
subarea_in_areas

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   9.3267e-05 ,   0.00285251 ,   0.00294578
```

```
     30 ,  0.000160588 ,   0.00624193 ,   0.00640252
    100 ,  0.000440777 ,   0.00951856 ,   0.00995934
    300 ,   0.00124398 ,    0.0115379 ,    0.0127819
   1000 ,   0.00408962 ,    0.0132879 ,    0.0173775
   3000 ,    0.0143013 ,    0.0146676 ,     0.028969
  10000 ,    0.0447937 ,    0.0192539 ,    0.0640476
  30000 ,     0.154985 ,    0.0218035 ,     0.176788
 100000 ,      0.53837 ,    0.0241263 ,     0.562496
 300000 ,      1.75548 ,     0.022996 ,      1.77848
1000000 ,      5.86824 ,    0.0546848 ,      5.92293
>
** End of commands from 'perftest-subareas.txt'
>
** End of commands from 'perftest-compulsory.txt'
> read "perftest-all_subareas.txt"
** Commands from 'perftest-all_subareas.txt'
> # Test the performance of subarea containment
> perftest all_subareas_in_area 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:
all_subareas_in_area

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,   9.2946e-05 ,   0.00346513 ,   0.00355807
     30 ,  0.000162413 ,   0.00950649 ,    0.0096689
    100 ,  0.000504737 ,    0.0172646 ,    0.0177693
    300 ,   0.00135195 ,    0.0261102 ,    0.0274622
   1000 ,   0.00424187 ,    0.0414083 ,    0.0456502
   3000 ,    0.0137365 ,     0.057978 ,    0.0717145
  10000 ,    0.0517152 ,    0.0798399 ,     0.131555
  30000 ,     0.156609 ,    0.0921668 ,     0.248776
 100000 ,     0.534243 ,    0.0703921 ,     0.604635
 300000 ,      1.96657 ,     0.118942 ,      2.08551
1000000 ,      6.53301 ,    0.0860047 ,      6.61902
>
** End of commands from 'perftest-all_subareas.txt'
> read "perftest-places_closest_to.txt"
** Commands from 'perftest-places_closest_to.txt'
> # Test the performance of places_closest_to
> perftest places_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:
places_closest_to

     N ,   add (sec) , cmds (sec) , total (sec)
    10 ,  0.00011462 , 0.00147599 ,  0.00159061
    30 , 0.000237863 , 0.00380161 ,  0.00403947
   100 , 0.000709101 ,  0.0063734 ,   0.0070825
   300 ,  0.00209734 ,   0.012326 ,   0.0144234
```

```
  1000 ,  0.00566928 ,  0.0204073 ,   0.0260766
  3000 ,   0.0128422 ,  0.0518677 ,   0.0647099
 10000 ,   0.0455383 ,   0.193328 ,    0.238866
 30000 ,    0.148753 ,   0.915244 ,       1.064
100000 ,    0.606475 ,    3.63841 ,     4.24489
300000 ,     2.03171 ,    12.5651 ,     14.5968
>
** End of commands from 'perftest-places_closest_to.txt'
> read "perftest-remove.txt"
** Commands from 'perftest-remove.txt'
> # Test the performance of removal
> perftest remove_place 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_place

      N ,   add (sec) , cmds (sec) , total (sec)
     10 , 0.000192446 ,  0.0260547 ,  0.0262471
     30 , 0.000368261 , 0.00898466 , 0.00935292
    100 , 0.000837344 ,   0.008958 , 0.00979535
    300 ,  0.00175754 ,  0.0111397 ,  0.0128972
   1000 ,  0.00638622 , 0.00850193 ,  0.0148882
   3000 ,   0.0278783 ,  0.0205953 ,  0.0484736
  10000 ,   0.0845412 ,  0.0426427 ,   0.127184
  30000 ,    0.300625 ,  0.0513126 ,   0.351937
 100000 ,    0.988364 ,  0.0505712 ,    1.03893
 300000 ,     3.29406 ,  0.0755235 ,    3.36958
1000000 ,     11.6332 ,  0.0740275 ,    11.7072
>
** End of commands from 'perftest-remove.txt'
> read "perftest-common_area.txt"
** Commands from 'perftest-common_area.txt'
> # Test the performance of common area search
> perftest common_area_of_subareas 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:
common_area_of_subareas

      N ,    add (sec) ,   cmds (sec) ,  total (sec)
     10 ,  0.000121083 ,   0.00343487 ,   0.00355596
     30 ,  0.000190972 ,    0.0109644 ,    0.0111554
    100 ,  0.000542697 ,     0.021534 ,    0.0220767
    300 ,   0.00145512 ,      0.03296 ,    0.0344152
   1000 ,     0.005265 ,    0.0469192 ,    0.0521842
   3000 ,    0.0153619 ,    0.0633264 ,    0.0786883
  10000 ,    0.0552876 ,    0.0810312 ,     0.136319
  30000 ,      0.19441 ,    0.0986835 ,     0.293094
 100000 ,     0.725671 ,     0.102702 ,     0.828374
 300000 ,      1.99309 ,     0.169423 ,      2.16251
```

```
1000000 ,       6.2709 ,     0.140803 ,      6.41171
>
** End of commands from 'perftest-common_area.txt'
>
** End of commands from 'perftest-all.txt'
```