==== Data Creation ====

Παράδειγμα εκτέλεσης: python3 genData.py -k keyFile.txt -n 100 -d 3 -l 4 -m 5

Αρχικά, το max level του nested string που παίρνουμε ως argument είναι τα max (pairs key -> value) , εκτός του high key

Επίσης, στη περίπτωση που δωθεί max_number_of_keys μεγάλυτερο απο το length που έχει το keyfile.txt,τότε το max_number_of_keys τότε γίνει ίσο με το length του keyfile.txt

Οσον Αφόρά το dataset η υλοποιηση γίνεται με αναδρομή (def generate_nested_string).

==========

==== Key Value Store ====

Παράδειγμα εκτέλεσης client: python3 kvClient.py -s serverFile.txt -i dataToIndex.txt -k 2

Παράδειγμα εκτέλεσης server1: python3 server1.py -a 127.0.0.1 -p 5001

Παράδειγμα εκτέλεσης server2: python3 server2.py -a 127.0.0.1 -p 5002

Παράδειγμα εκτέλεσης server3: python3 server3.py -a 127.0.0.1 -p 5003

Εχουν υλοποιηθεί όλες οι λειτουργίες.

Αρχικά, απο τη πλευρά του client γινεται conection με όλους του server με sockets και περίπτωση που δεν συνδεθεί με κάποιον τερματίζει.

Αφου συνδεθεί, στέλνει random σε k server κάθε record, και αφου τελειώσει ρωτάει τους servers αν είχαν κάποιο error κατα την αποθηκευση των records.

(Αυτο η το check απο τη πλευρά των server γίνεται με μία Boolean μεταβλητή, οπου γίνεται false στη περίπτωση που δεν γίνει επιτυχώς insert κάποιο record.

Η αποθήκευση γίνεται σε trie με βάση μόνο το high key.

Μετα απο κάθε ερώτηση του χρήστη, πρίν στείλει ο client το ερώτημα τσεκάρει αν έχουμε τουλάχιστον k server UP, αν όχι τερματίζει.

Ο κάθε server αναγνωρίζει τι τύπος ερώτησης είναι καθώς στο μήνυμα συμπεριλαμβάνεται και το GET, QUERY κλπ.

a) GET key: στέλνει το ερώτημα σ' όλους στους server και αυτό στη συνέχεια καλούν την trie.find οπου επιστρέχει το value.

Στη περίπτωση που δεν υπάρχει επιστρέφει NOT FOUND

b) DELETE key: στέλνει το ερώτημα σ' όλους τους server, και ο καθένας διαγράφη trie.delete

c) QUERY keypath: στέλνουμε ολο το keypath σ' όλους τους server, οπου ο κάθε server κάνει split το πρώτο key -> value εκτελεί την trie.find να να βρεί το value

και στη συνέχεια καλεί την search οπου παίρνει ο είσοδο όλα τα keys και το αρχικό value. και επιστρέψει το value που αναζητάμε αν το βρεί.

Η search μετατρέπει τον keypath σε dictionary για να γίνει πιο ευκολη η αναζήτηση και το επιστρέφει στην μορφή που ήταν αρχικά.

d) COMPUTE f(x) WHERE x = QUERY key1.key2…: Αρχικά ο client κάνει split ολα τα queries και στα στέλνει σ όλους τους server. αφού πάρει τις απαντήσεις τότε υπολογίζεται η συνάρτηση.

***Στο dataset τα key και τα values είναι με "". Ενώ στα question είναι χωρίς!

Παράδειγμα Εκτέλεσης:

$ python3 genData.py -k keyFile.txt -n 100 -d 3 -l 4 -m 5

Εγινε η δημιουργία 100 records αρχείο, το οποίο υπάρχει και στο repository

$ python3 server1.py -a 127.0.0.1 -p 5001

$ python3 server2.py -a 127.0.0.1 -p 5002

$ python3 server3.py -a 127.0.0.1 -p 5003

$ python3 kvClient.py -s serverFile.txt -i dataToIndex.txt -k 2

Start reading arguments

End reading arguments

Server: 127.0.0.1 : 5001 IS UP!

Server: 127.0.0.1 : 5002 IS UP!

Server: 127.0.0.1 : 5003 IS UP!

All Servers is UP

Start send data to index in Servers

End send data to index in Servers

Index Data: OK

Index Data: OK

Index Data: OK

Enter Question: GET key1

Received Answer: key1 -> ["level" -> ["height" -> 80.38 | "name" -> "oq" | "level" -> 26] | "height" -> ["name" -> "uk" | "age" -> 18 | "level" -> 89 | "height" -> 10.56 | "street" -> "fo"] | "age" -> ["street" -> "r" | "height" -> 59.67 | "age" -> 45 | "level" -> 7] | "street" -> [] | "name" -> ["street" -> "fopv" | "height" -> 43.01]]

Enter Question: GET key1000

Received Answer: key1000 -> NOT FOUND

Enter Question: DELETE key1

Received Answer: key1 -> OK

Enter Question: GET key1

Received Answer: key1 -> NOT FOUND

Enter Question: QUERY key2.street

Received Answer: key2.street -> ["height" -> 74.13 | "name" -> "v"]

Enter Question: QUERY key2.street.name

Received Answer: key2.street.name -> v

Enter Question: QUERY key2.street.nam

Received Answer: key2.street.nam -> NOT FOUND

Enter Question: COMPUTE 2*x WHERE x = QUERY key3.height.level

Received Answer: 108

Enter Question: COMPUTE log(x)+tan(y)-2*x WHERE x = QUERY key3.height.level AND y = QUERY key10.age.level

Received Answer: -332.21845269437216

Enter Question: GETT key3

Wrong format of question! Try again

Enter Question: COMPUTE 2*x WHERE x = QUERY key3.height.name

ERROR: x = QUERY key3.height.name is STRING, can not compute

Enter Question: COMPUTE 2*x WHERE x = QUERY key3.height.id

ERROR: x = QUERY key3.height.id NOT FOUND

Enter Question: COMPUTE log(x)+tan(y)-2*x WHERE x = QUERY key3.height.level AND y = QUERY key10.age.le

ERROR: y = QUERY key10.age.le NOT FOUND

#No we test with down servers, In our test we have 3 total servers and 2 more index

#We down one server

Enter Question: COMPUTE log(x)+tan(y)-2*x WHERE x = QUERY key3.height.level AND y = QUERY key10.age.level

Received Answer: -332.21845269437216

#And now we down onother one

Enter Question: COMPUTE log(x)+tan(y)-2*x WHERE x = QUERY key3.height.level AND y = QUERY key10.age.level

WARNING: All server are= 2 or more servers are down and therefore it cannot guarantee the correct output

ERROR: x = QUERY key3.height.level NOT FOUND

#we down and the last server

Enter Question: COMPUTE log(x)+tan(y)-2*x WHERE x = QUERY key3.height.level AND y = QUERY key10.age.level

All server are down