**Q1**

Acknowledgements

* Data was obtained from <https://statecancerprofiles.cancer.gov/incidencerates/index.php>
* Help generously given by amazing TF Wenxin and Professor McDougal

Data Cleaning

* All descriptors for the data are deleted, including everything except: column headings, state names, and data;
* All data columns that are not state names and age-adjusted incidence rates are deleted as they are irrelevant to the assignment;
* All “(#)” removed from state names;
* The age-adjusted incident rates column was renamed ‘Age-Adjusted IR’ to allow easier manipulation in code; and
* The state names were all changed to lower case such that, by using name.lower() with all received input names from the website, the input would be able to match to the dataframe’s “State” column values despite capitalization of input string.

Validity checks of state name: if the state name entered, despite capitalization, did not match any of the state names within the dataframe, then it is considered invalid. For example, typing errors would lead the user to a failure.html page that says: “You have failed to access state information because the name you entered was invalid!”

Extension

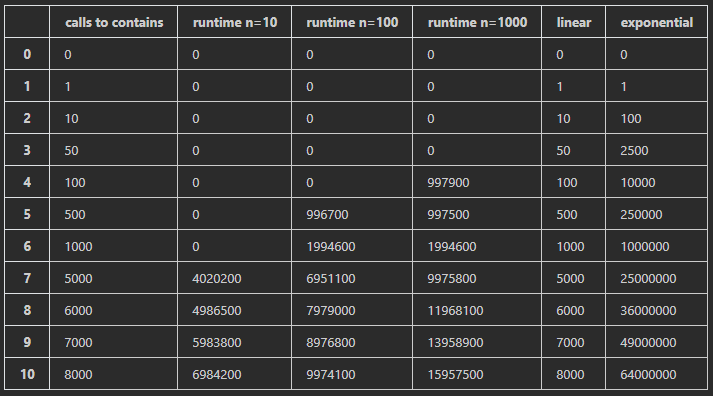
I added a drop-down menu on the index page where users can choose to either 1) enter the state name manually or 2) choose from existing choices. Both would lead to the information being sent to /info and then having the state name displayed. I also disabled both entering a state name and choosing one such that duplicate information would not be sent to /info.

**Q2:**

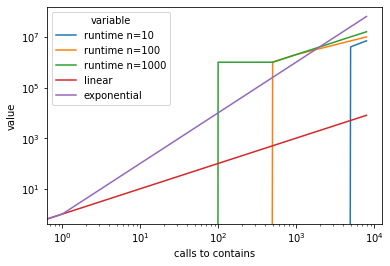
Code, output, and tests are shown in later pages under “**Code and Output on Following Pages**.”

Note that t\_1, t\_2, and t\_3 refers to trees with varying sizes of 10, 100, and 1000 respectively. This could be altered as the size of trees with random values generated could be decided by inputting the value in the generate\_trees function parameters. This also applies to how many times the range of random numbers to be generated and added to the trees in calc\_runtime function.

The table and graphs of runtimes are shown below.



Initially, only the values 0, 1, 10, 100, and 1000 were tested (number of calls to contains (in)), but it seems that most initial runtime values are 0 hence the graph is not very informative, so 0, 1, 10, 50, 100, 500, 1000, 5000, 6000, 7000, and 8000 are the final tested values. Note that with each run of populating trees of different sizes, the runtime is different, shown below is a result specifically picked such that the runtimes become non-zero at different number of calls to contains, such that the graphs do not have two lines overlapping each other and then branching out separately, which could be confusing. Linear and exponential columns were added to the and graphs for comparison with runtime results.



Note that for the log-log graph above , all runtime values for checking “in” (except linear and exponential) eventually fall within the range of linear and exponential, hence runtime is O(logn).

Chart, line chart

Description automatically generated

Note that for the log-log graph above, the runtime for setting up trees with 0, 1, 10, 50, 100, 500, 1000, 5000, 6000, 7000, and 8000 randomly generated nodes are shown in blue, compared to linear and exponential lines. The blue line eventually falls within the range of linear and exponential lines.

**Q3:**

I was not able to produce any deliverable for Q3. The concept of a quad tree is too unfamiliar to me despite discussing in lecture and in the video provided. I could not understand how one should initialize cluster centers and determine which box belongs to which center point. A tree is hierarchical, but KNN neighbors (and their centers) are not, so I do not understand how this could be represented in a tree format. Moreover, the recursion required to add non-center nodes to the tree is also confusing to me. I don’t understand the calculation required to determine which node belongs to which subtree, because I do not understand how boxes could be represented by these trees. I tried writing some code, but they are not complete and most likely wrong down to the fundamental logic. If this exercise could be discussed extensively in class, that would be greatly appreciated!

**References**

National Cancer Institute. (2021, December 3). Incidence Rates Table. State Cancer Profiles. chart. Retrieved December 3, 2021, from https://statecancerprofiles.cancer.gov/incidencerates/index.php?stateFIPS=00&amp;areatype=state&amp;cancer=001&amp;race=00&amp;sex=0&amp;age=001&amp;stage=999&amp;year=0&amp;type=incd&amp;sortVariableName=rate&amp;sortOrder=default&amp;output=0#results.

<https://stackoverflow.com/questions/55447599/how-to-send-data-in-flask-to-another-page>

**Code and Output on Following Pages**