LeetCode Submission

Jordan Halaby

8/26/18

Problem Chosen

The problem that I chose to solve (on medium difficulty with no official solution) was the “Kth smallest element in a sorted matrix” via Python3. The goal of the problem was: “Given a n x n matrix where each of the rows and columns are sorted in ascending order, find the kth smallest element in the matrix. Note that it is the kth smallest element in the sorted order, not the kth distinct element”.

My Solution

I chose to use Python3 for this problem. My mindset going into the problem was to convert the matrix, which was essentially a two-dimensional list, into a single-dimension list. After the matrix was converted, I decided to sort the list and get the element at the kth – 1 index. The reason for using the kth – 1 index was that the matrix starts at location 1,1 which is problematic because list indexing in Python3 starts at 0.

A picture of my code submission

A screenshot of a cell phone

Description automatically generated

Runtime and Space Complexity Analysis

Since my solution to the problem was so simple and terse, I was able to obtain exceptional results for my runtime statistics. A picture of the complexity analysis is displayed in the next section. My runtime was 200 milliseconds (which is approximately 0.2 seconds), which was faster than 84.5% of online submissions for the same problem. My memory usage on the other hand didn’t have as good of results compared to other user submissions. I used 19.9 megabytes (which is approximately 0.0199 gigabytes); compared to others this memory usage was only less than 9.09% of other user submissions. I think that my runtime was so fast because my function script was so short, even though I did sort the array; the memory usage was so high because I stored the full matrix into a list, so the matrix and the new list both existed in memory.

A picture of my runtime and space complexity outcome

A picture containing animal

Description automatically generated

Conclusion

This was a very simple problem to solve because Python3 has built-in functionality to create and manage single and two-dimensional lists. The key to solving this problem was to sort the converted matrix so that the kth element could be found with simple list indexing.