Assignment Review

The assigned task involved conducting market research on the Play Store, addressing specific queries related to app categories, top companies, prolific developers, app pricing, and download statistics. The objective was to process a large dataset and provide insightful answers to aid marketing decisions. I utilized distinct functions to address each query, enabling code modularity and readability. For most queries, I employed separate functions for parsing the dataset, as well as functions for writing the necessary information to a separate file. However, for one query, I used a single function to perform both tasks. This coding method allowed me to test and debug my code more efficiently, without the need to run the entire code each time.

One of the significant challenges encountered was handling inconsistencies and anomalies within the dataset. To tackle this issue, I first worked with a "mock file" that I created by taking about 1000 lines from the original file. Once I got everything to work on that file, I thought I was done with the assignment and that there would not be any more problems with the large file. However, the regex I initially wrote did not cover all the inconsistencies inside the file, making reading the file and parsing it correctly the most challenging task of the whole assignment. It took me a lot of time to successfully implement the right regex, which was something I did not expect to be so challenging.

I initially thought that the second task, where we needed to create a file containing the top 100 companies with the most applications, would be the hardest part of the assignment. I assumed that we needed to find the exact name of the company inside the App ID, and not just the first two words regardless of whether it was a name or two domains. However, the task turned out to be quite easy to figure out and complete.

If I were to do the assignment again, I would most likely do it similarly. Maybe one of the better ways of optimizing memory usage is to process the code in smaller batches and then process the code. The main issue I encountered was the high memory consumption of the list that contained all the lines from the entire file. This was a side effect of using a single function to read the entire file. I believe that there are also some other implementations better than mine that could be done to improve the efficiency of the program. Overall, I think my implementation is good and understandable.

Tarik Bašić