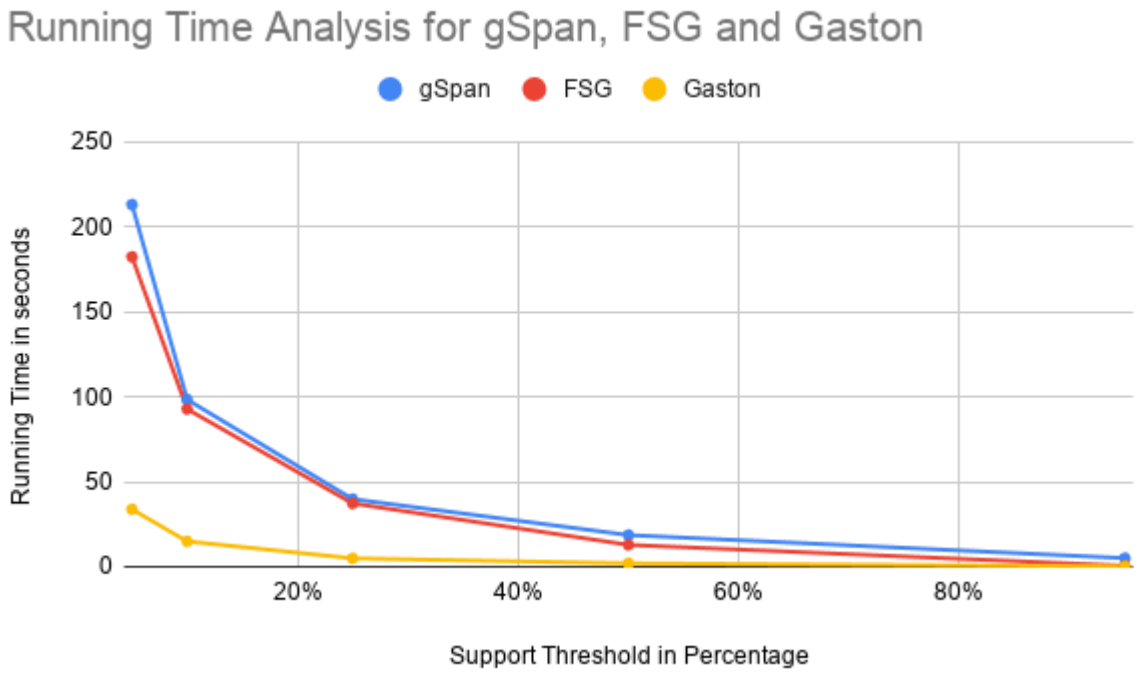
- The plots obtained corresponding to the different algorithms can be found below:



- Here as we can see, for higher values of support threshold; the time taken by these different methods will get lesser.
- This is because with decreasing minsup, the number of frequent itemsets generated will be very high. Thus resulting in such large time values. The growth rate seems to be decreasing here with higher thresholds. And this should be because of the fact that the number of frequent itemsets increase exponentionally with decrease in the threshold value. And so, for higher threshold values, the growth rate becomes stagnant
- Here, we can see that the time returned by the Gaston method is much lesser than the time taken by the other two methods. The reason behind this is that Gaston is a highly optimised program which makes the use of Embedded list structures within itself that would lead to improving the time required for finding the sub-graph isomorphism[3]. Thus, because of this cluster type structure utilised in the Gaston algorithm, the time taken to check the isomorphism will be lower as compared to the other algorithms which go over the many combinations for the same. And since at lower thresholds there will be many frequent itemsets, and thus many cases to check for isomorphism; the performance of the gaston algorithm is much better in terms of time than the other two methods for lower support thresholds.
- If we look at the other two methods of FSG and gSpan then from the content taught to us in the lectures; it would be deduced that the performance of FSG should be somewhat lesser than the performance of gSpan algorithm. Because FSG performs a BFS while gSpan performs DFS inherently. Such experimental evidences were even described in the gSpan paper[1]. In the last release of the FSG paper[2], the authors do talk about the gSpan algorithm and the performance comparison between them where, especially for synthetic datasets, FSG is reported to perform better.
- Thus we can say that because of some particular structural properties of the datastructure used by us, the performance of FSG and gSpan are almost the same; with FSG being slightly better than the gSpan algorithm. It seems that the dataset used by us would not be similar to the synthetic datasets used by the gSpan algorithm where they are found to be more performant than the other standard methods

References

[1] Yan, Xifeng, and Jiawei Han. "gspan: Graph-based substructure pattern mining." 2002 IEEE International Conference on Data Mining, 2002. Proceedings.. IEEE, 2002.

[2] Kuramochi, Michihiro, and George Karypis. "An efficient algorithm for discovering frequent subgraphs." *IEEE Transactions on Knowledge and Data Engineering* 16.9 (2004): 1038-1051.

[3] Nijssen, Siegfried, and Joost N. Kok. "The gaston tool for frequent subgraph mining." Electronic Notes in Theoretical Computer Science 127.1 (2005): 77-87.