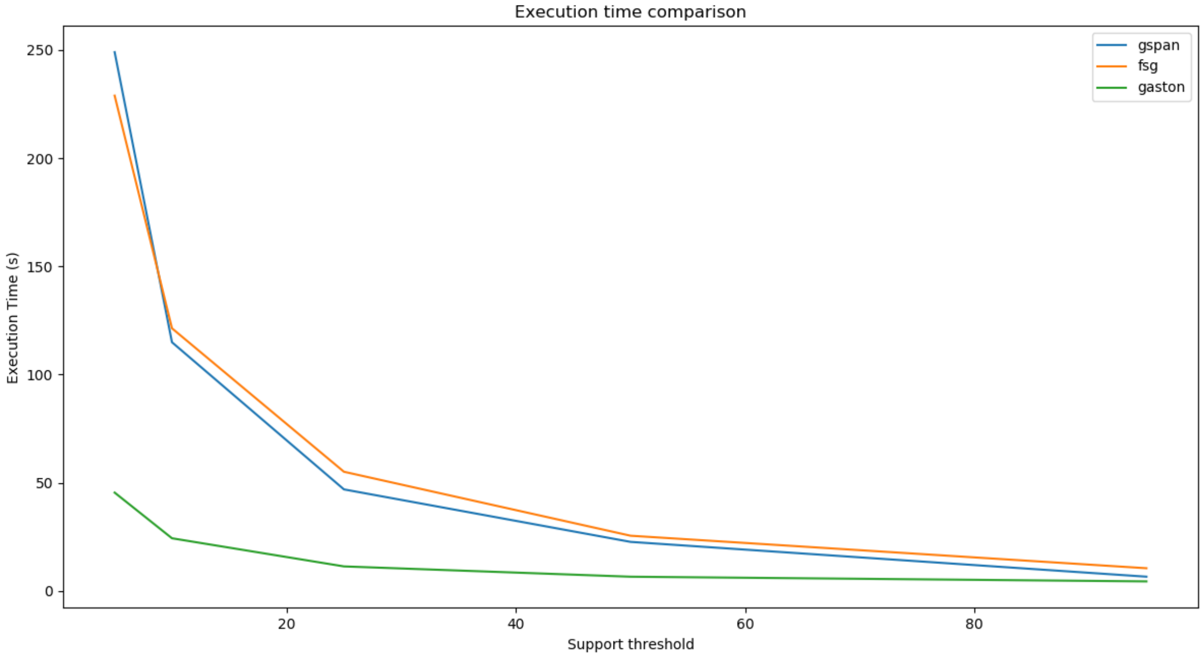
COL761: Data Mining
Assignment 3 : Report

Implementations of gSpan, FSG and Gaston were run on the AIDS dataset at minSups = 5%, 10%, 25%, 50% and 95%. Their running times were plotted and are shown in the following graph:



Observations:
We find that the running time of each of the algorithms decreases exponentially as the minSup is increased. This is because when the minSup is increased, the number of subgraphs crossing the threshold decreases at each step of the algorithms, thus reducing the overall runtime significantly.
It is also observed that in terms of run times,

FSG > gSpan > Gaston

FSG has the highest run time. FSG is a join-based (BFS) approach which produces all possible candidates at a level and then prunes them after checking for the required conditions. This procedure of redundant candidate generation is very computationally expensive.
gSpan shows improvement over FSG as it eliminates redundant candidate generation and pruning by using minimal DFS codes. Moreover, at each iteration, the mining procedure is performed in a way that the whole graph dataset is shrunk to the one containing a smaller set of graphs, with each having less edges and vertices, thus giving it a competitive edge over FSG.

Gaston has the fastest run time. The Gaston algorithm keeps all embeddings (mapping of nodes and edges of a subgraph to corresponding nodes and edges in the graph in which it occurs) in order to generate only refinements which actually appear and to generate fast isomorphism testing. By considering fragments that are paths or trees first, and by only proceeding to general graphs with cycles at the end, a large fraction of the work can be done efficiently. For the subgraph isomorphism problem, Gaston defines a global order on the cycle-closing edges and only generates those cycles that are *“larger”* that the last one. Detection of duplicates is achieved in two steps: in step one hashing is done to pre-sort and in second step a graph isomorphism test is done in order to find the final duplicate. (Ref. : A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston, Marc Wo¨rlein, Thorsten Meinl, Ingrid Fischer, and Michael Philippsen https://www2.informatik.uni-erlangen.de/publication/download/PKDD05.pdf)