

Bonus Mark Enhancements

To speed up our FP-Growth algorithm we made an enhancement when projecting the subtrees of our global tree. In our ProjectTrees class we implemented an efficient method to deal with merging branches, which is done in mergeBranches(). This method does this by iterating through all the nodes in the tree and checking if they are a “candidate to merge”, meaning, they have more than 2 children. We then iterate through the children which will be typically 1-3 nodes (rarely more), and comparing all of them against eachother. This would be worst case $O(n*n)$ but in practice we build the tree so that it does not have any nodes with $O(n)$ number of children. Now, once the children are merged we have to trigger a flag (changesMade) to true and try again, which again, unless there is a terrible made tree with nodes with $O(n)$ children this will be quite rare. Ignoring these cascading merges would have resulted in a incorrect result.

Unfortunately, we ran out of time to implement the projections beyond depth 2, hence our results are wrong. We can correctly mine frequent patterns of size 2 but not beyond that. If we had managed our time better we would have implemented a way to make the “buildProjectedTree()” recursive and work well. The pro of having an incorrect result is that our algorithm is fast!! We here at Artificial Stupidity believe that is worth some marks :D