Project 2 Branch Predictor

Noah Schrock

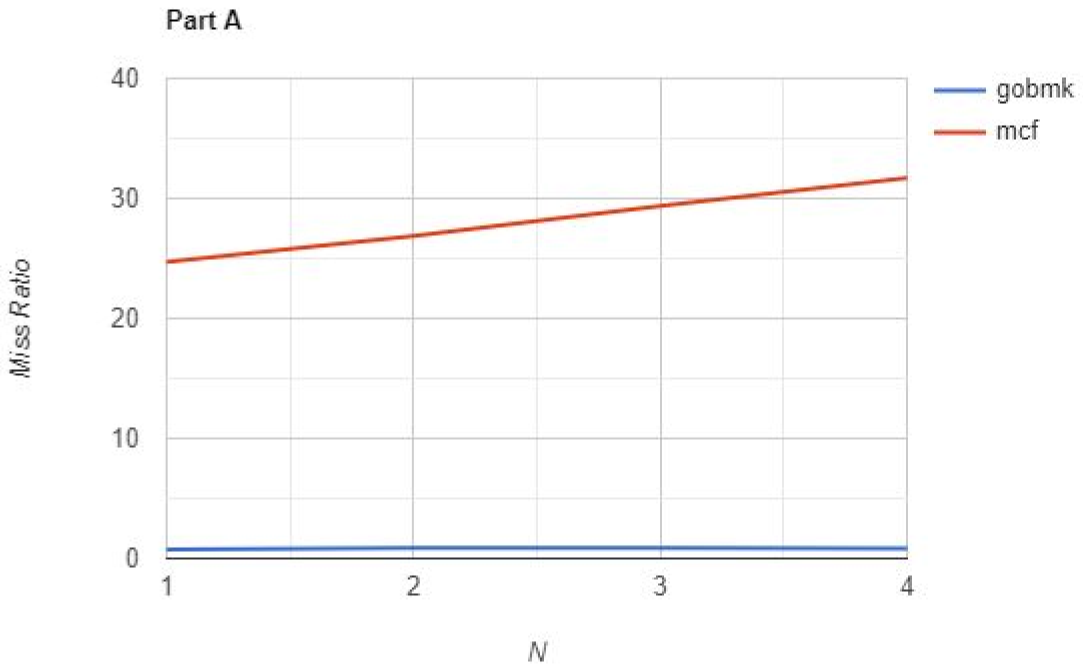
EEL 4768

11/20/20

How to run my code in Visual Studio Code

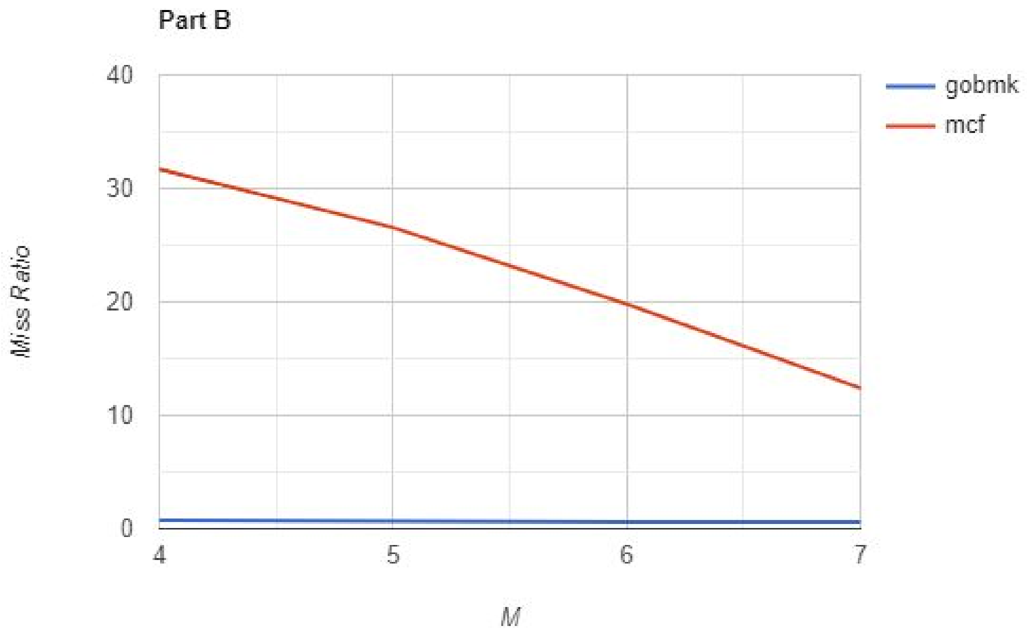
1) Open the Branch Predictor folder in the workspace
2) Go to Terminal at the top, and then hit run build task…
3) Chose “C/C++: g++.exe build active file” from the drop down menu
4) Hit terminal again, and choose new terminal
5) In the terminal type “g++ -o SIM main.cpp”
6) Now you can begin runs using the format “./sim M N trace_file.txt”

Part A



Part A was different from the rest of the runs, this is because our N value increased. The value of N increases by 1 for each run and the mcf trace file followed this trend by having an almost linear increasing value for the miss ratio.This was the opposite of the rest of the runs where we increased the M value and the miss ratio decreased. This part had the worst accuracy, or highest miss ratio, of all of the parts we did. It had an average miss ratio of .83% for the gobmk trace file and an average miss ratio of 28.16% for the mcf trace file.

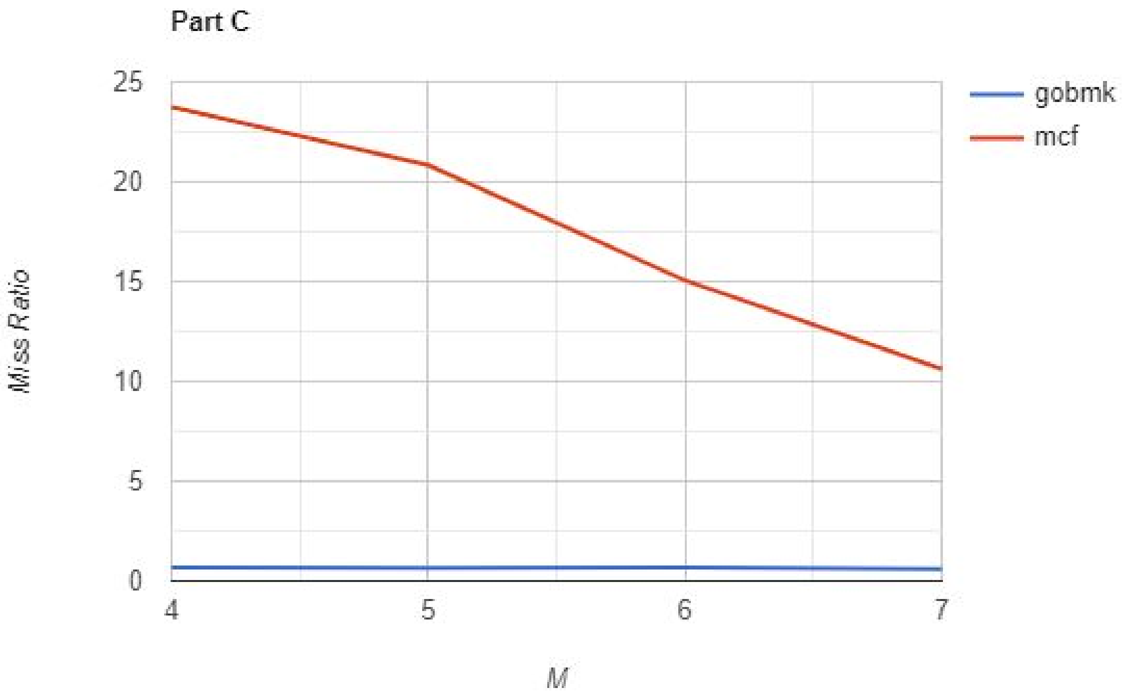
Part B



Part B was the most average of all the parts we had, it had the lowest value for gobmk trace file out of all the runs across all parts, it tied part C with the overall miss rate for to gobmk trace file with a .66% miss rate.

The miss rate for the mcf file was 22.62% average using the data from all 4 runs. The trend shows the miss ratio decreasing rapidly as the value of M increases, this is similar to part B but the opposite of part A. Part B tied with part A for the largest miss rate on a single run with 31.72% since it used the same values as the last run, but the over all miss rate was a lot lower for part B than for Part A

Part C



The gobmk file had a variation of .09% which is much less than the .19% of part B, both have a very small difference, but the one from part C is the smallest of all the runs we did. Both part C and part B had decreasing miss rates as the M value increased from 4 to 7. Both of them ended up with the same average accuracy across the entire run, but part B had the lowest miss rate of any one individual run with .58%

The mcf file had a mich better miss rate in part C than it did in part B. The overall variation in miss rate was 13.13% which was a really big improvement from the 19.32% of part B. As the M increases from 4 to 7 the miss ratio decreases at a rapid rate, this was the trend for both parts. Part C had much better accuracy than part C with an average miss rate of 17.57% compared to part B's 22.62%