The State University of New York at Binghamton
Department of Computer Science
CS 520 – Spring 2019

Project #1: Branch Prediction

By

Madhumita Ghosal

Honor Pledge: I have neither given nor received unauthorized aid on this test or assignment. Student's electronic signature: __Madhumita Ghosal_____ (sign by typing your name)

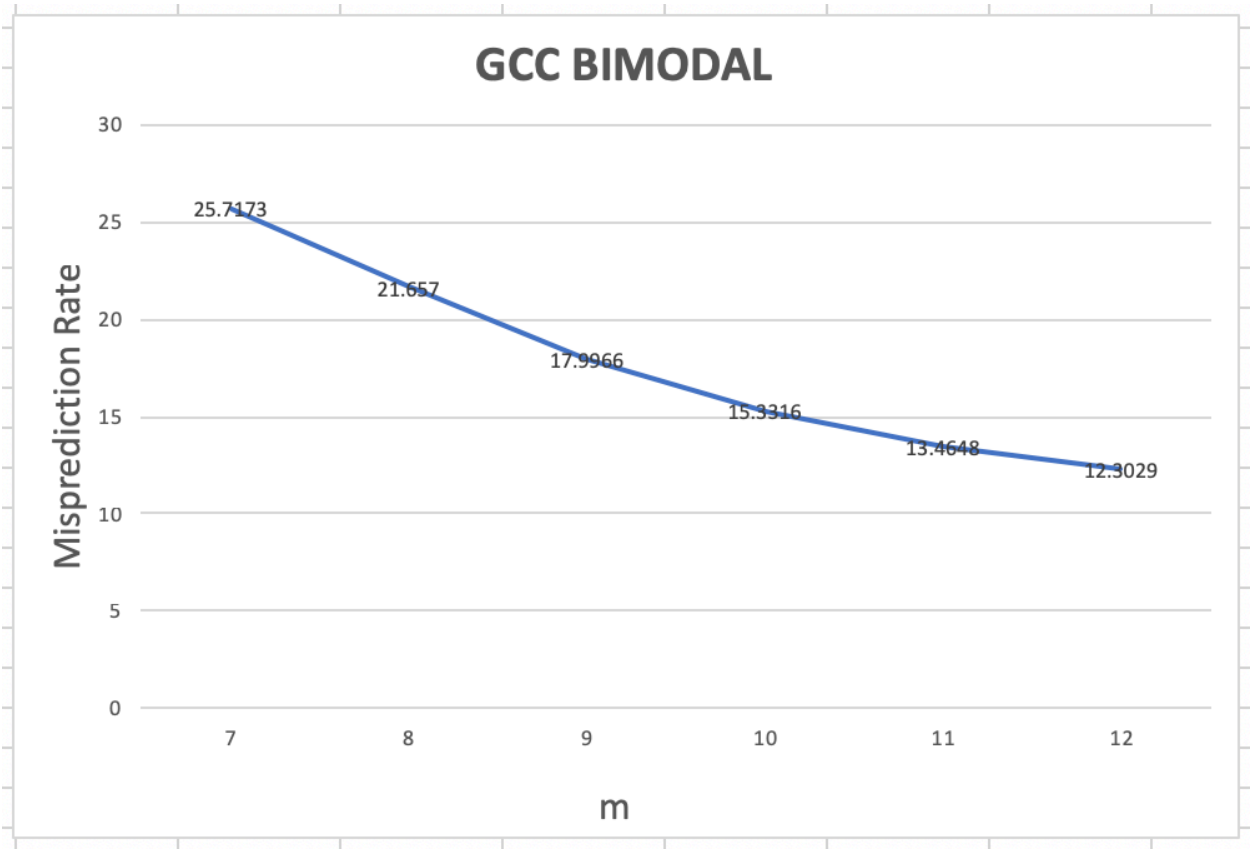
PART 1 - Bimodal Predictor

B)

Graph

- When the benchmark is GCC

m	Misprediction rate
7	25.7173
8	21.657
9	17.9966
10	15.3316
11	13.4648
12	12.3029

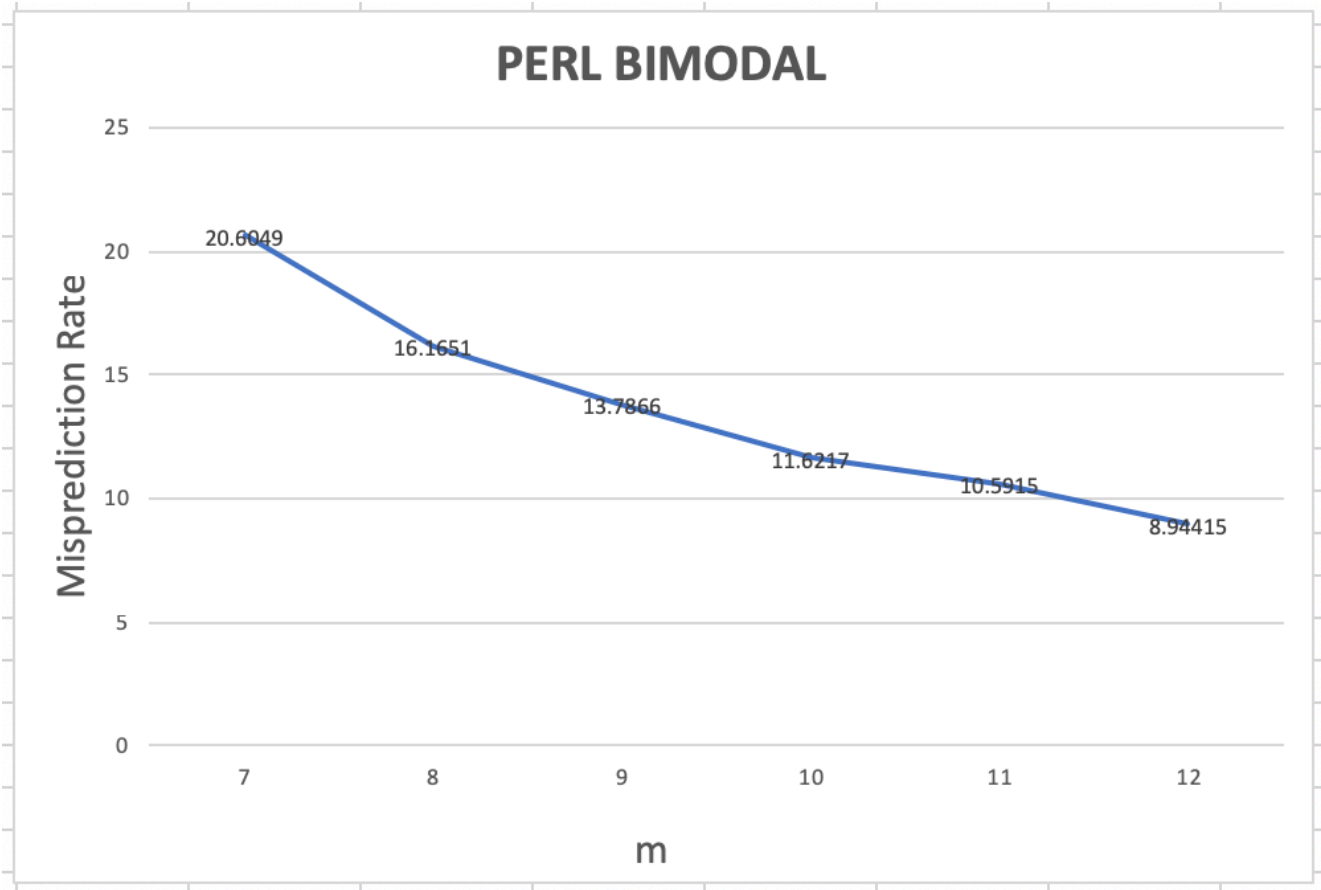


- When the benchmark is JPEG

m	Misprediction rate
7	7.84725
8	7.7476
9	7.69205
10	7.6523
11	7.51485
12	7.48835

- When the benchmark is PERL

m	Misprediction rate
7	20.6049
8	16.1651
9	13.7866
10	11.6217
11	10.5915
12	8.94415



Analysis

m	GCC	JPEG	PERL
7	25.7173	7.84725	20.6049
8	21.657	7.7476	16.1651
9	17.9966	7.69205	13.7866
10	15.3316	7.6523	11.6217
11	13.4648	7.51485	10.5915
12	12.3029	7.48835	8.94415

1. For each benchmark ie, GCC, JPEG and PERL, the misprediction rate is different even though the value of m is same.
2. One of the key analysis points is that with the increasing value of m, misprediction rate decreases.
3. For higher value of m, misprediction rate is the lowest. For smaller value of m multiple branches share the same counter value because of which misprediction rate is higher.
4. GCC benchmark shows a variation of ~ 12% to 26% when m ranges from 7 to 12.
5. JPEG benchmark does not show much variations in the misprediction rates.
6. PERL benchmark has a misprediction rate lower than gcc benchmark for the same value of m and the range of misprediction rate is also similar for both of them.

Design

m	GCC	JPEG	PERL
7	25.7173	7.84725	20.6049
8	21.657	7.7476	16.1651
9	17.9966	7.69205	13.7866
10	15.3316	7.6523	11.6217
11	13.4648	7.51485	10.5915
12	12.3029	7.48835	8.94415

With 16kB budget constraint, we can get minimum misprediction rate and minimum predictor cost by

a) Gcc - 15.3316% , Jpeg - 7.69205%, perl - 11.6217%

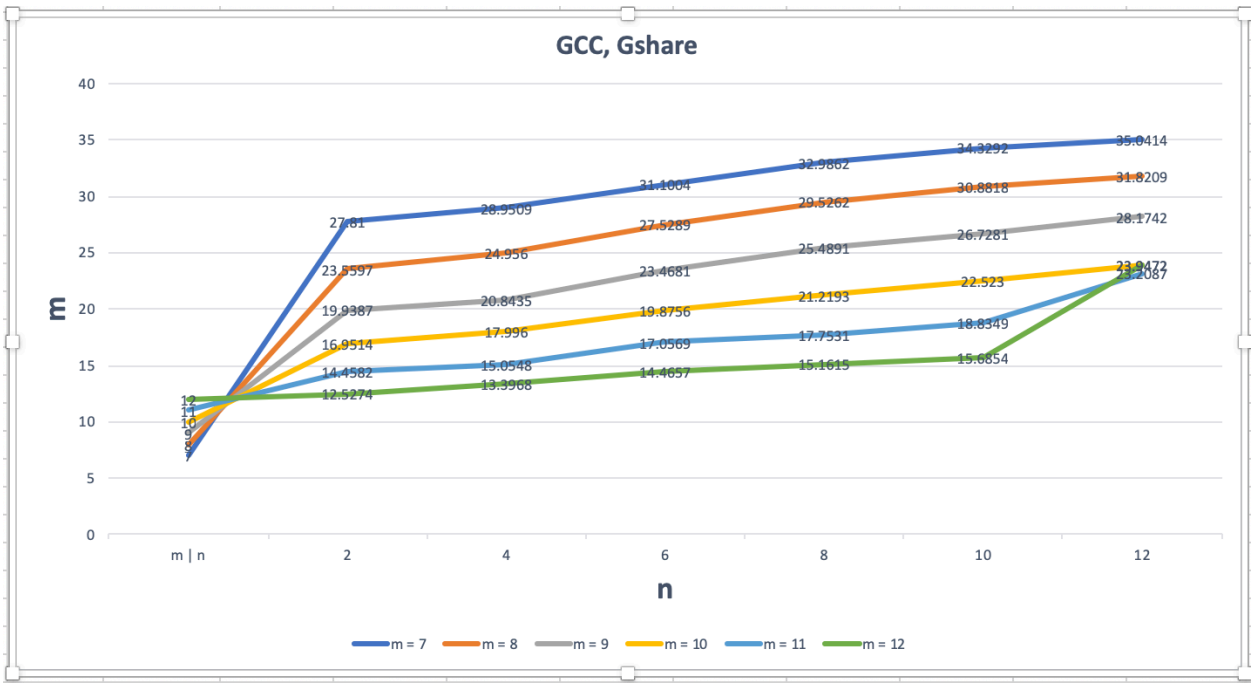
PART 2 - GShare Predictor

B)

Graph

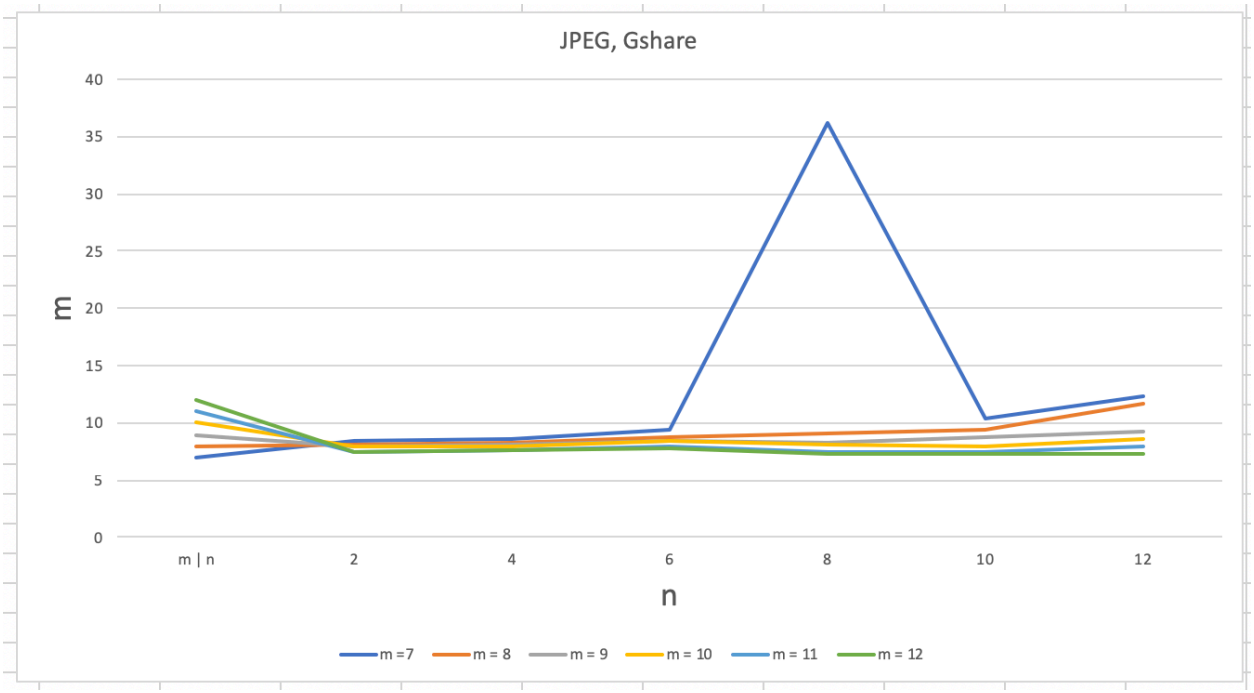
- When the benchmark is GCC

m | n	2	4	6	8	10	12
7	27.81	28.9509	31.1004	32.9862	34.3292	35.0414
8	23.5597	24.956	27.5289	29.5262	30.8818	31.8209
9	19.9387	20.8435	23.4681	25.4891	26.7281	28.1742
10	16.9514	17.996	19.8756	21.2193	22.523	23.9472
11	14.4582	15.0548	17.0569	17.7531	18.8349	23.2087
12	12.5274	13.3968	14.4657	15.1615	15.6854	23.9472



- When the benchmark is JPEG

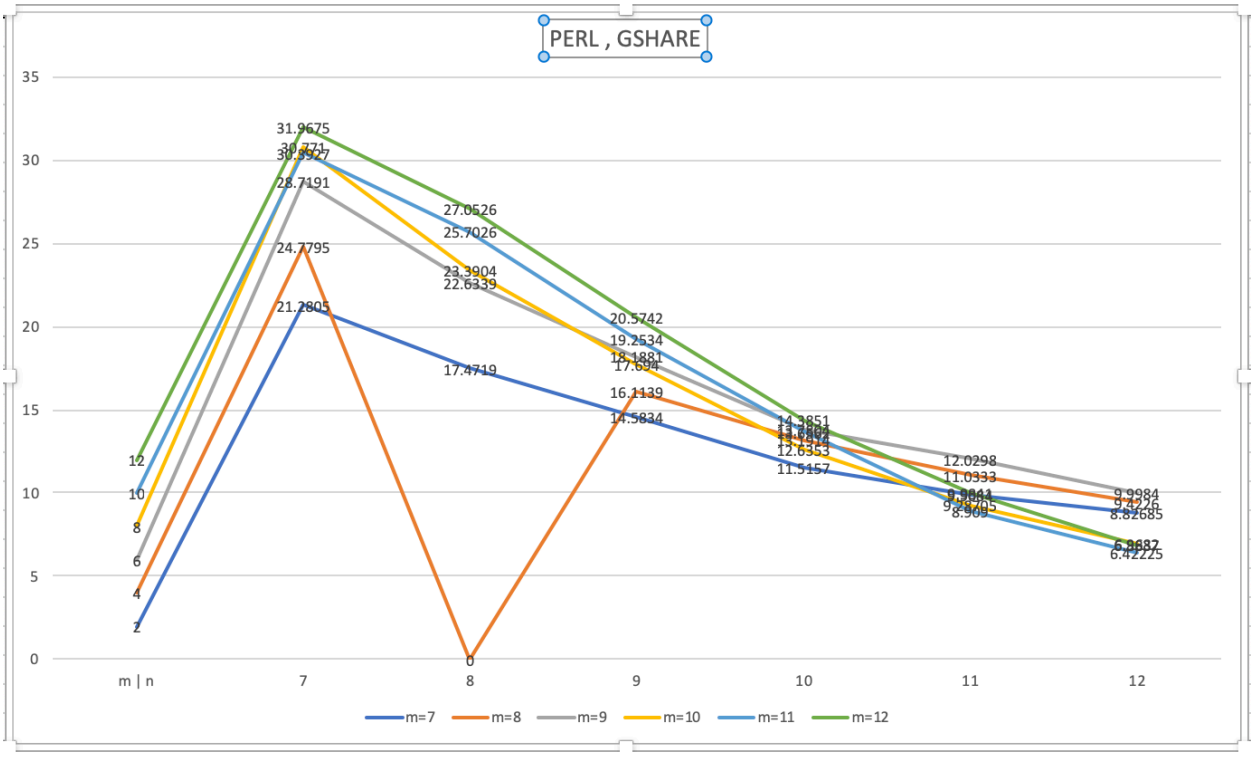
m | n	2	4	6	8	10	12
7	8.4056	8.6507	9.4929	36.2182	10.3598	12.2759
8	8.04005	8.19905	8.81785	9.06885	9.45385	11.6038
9	7.9398	8.0386	8.4578	8.3199	8.6878	9.1901
10	7.8712	7.96545	8.3731	8.0543	7.98455	8.6286
11	7.4532	7.61745	7.899	7.49775	7.4686	7.9425
12	7.43345	7.5997	7.8432	7.2759	7.2966	7.35235



- When the benchmark is PERL

m | n	2	4	6	8	10	12
7	21.2805	24.7795	28.7191	30.771	30.3927	31.9675
8	17.4719	19.0033	22.6339	23.3904	25.7026	27.0526
9	14.5834	16.1139	18.1881	17.694	19.2534	20.5742
10	11.5157	13.1914	13.7804	12.6353	13.6562	14.3851

11	9.9084	11.0333	12.0298	9.28705	8.909	9.9841
12	8.82685	9.4226	9.9984	6.9682	6.42225	6.8837



Analysis

1. As the value of m increases, misprediction rate decreases for all three benchmarks.
2. For higher values of m, misprediction rate initially high for lower value of n and it decreases with increased n and it increases again after a particular value of n.
3. For smaller values of n, branches indexed to the same entry in the gshare table detoriates the prediction.
4. For higher values of m, very less branches point to each index. Here, global history register is used to make the prediction better.

Design

- GCC

m | n	2	4	6	8	10	12
7	27.81	28.9509	31.1004	32.9862	34.3292	35.0414
8	23.5597	24.956	27.5289	29.5262	30.8818	31.8209
9	19.9387	20.8435	23.4681	25.4891	26.7281	28.1742
10	16.9514	17.996	19.8756	21.2193	22.523	23.9472
11	14.4582	15.0548	17.0569	17.7531	18.8349	23.2087
12	12.5274	13.3968	14.4657	15.1615	15.6854	23.9472

In my opinion, for gcc, m= 10 and n=4 gives best output between misprediction rate and minimum predictor cost.

- JPEG

m | n	2	4	6	8	10	12
7	8.4056	8.6507	9.4929	36.2182	10.3598	12.2759
8	8.04005	8.19905	8.81785	9.06885	9.45385	11.6038
9	7.9398	8.0386	8.4578	8.3199	8.6878	9.1901
10	7.8712	7.96545	8.3731	8.0543	7.98455	8.6286
11	7.4532	7.61745	7.899	7.49775	7.4686	7.9425
12	7.43345	7.5997	7.8432	7.2759	7.2966	7.35235

In my opinion, for gcc, m= 10 and n=8 gives best output between misprediction rate and minimum predictor cost.

- PERL

m | n	2	4	6	8	10	12
7	21.2805	24.7795	28.7191	30.771	30.3927	31.9675
8	17.4719	19.0033	22.6339	23.3904	25.7026	27.0526
9	14.5834	16.1139	18.1881	17.694	19.2534	20.5742
10	11.5157	13.1914	13.7804	12.6353	13.6562	14.3851
11	9.9084	11.0333	12.0298	9.28705	8.909	9.9841
12	8.82685	9.4226	9.9984	6.9682	6.42225	6.8837

In my opinion, for gcc, m= 11 and n=10 gives best output between misprediction rate and minimum predictor cost.