Questions are answered below:

1. Why and how varying the size of the PHT impacts (or not) the prediction accuracy for each of the predictors?
   Ans: Varying the size of the Pattern History Table (PHT) impacts prediction accuracy by altering the amount of historical branch behavior captured by the predictor. A larger PHT stores more historical information, allowing predictors like Gshare to recognize longer-term patterns and adapt to more complex behaviors. This will enhance accuracy, particularly for branches with extended histories and intricate dependencies.
2. What are the advantages and disadvantages of the Local predictor?
   Advantages:
   Precise predictions based on the unique history of each branch, making it effective for branches with distinct local patterns.
   The Local predictor typically has lower latency as it only relies on the history of the specific branch, allowing for quicker predictions.
   Disadvantages:
   Limited effectiveness for branches with global patterns, higher storage and access complexity due to separate histories.
   The Local predictor lacks a broader view of the program's execution patterns since it relies solely on the history of the individual branch. This can lead to suboptimal predictions for branches with complex or global patterns.

3. What are the advantages and disadvantages of the Gshare predictor?
   Advantages:
   Gshare combines global history with branch-specific information, enhancing prediction accuracy by capturing correlations between different branches. It accounts for both branch history and patterns seen across the program. Adaptability to Diverse Patterns:
   Gshare is effective in handling a wide range of branch behavior patterns, from short-term to long-term dependencies. This adaptability makes it suitable for various types of programs and code structures.
   Disadvantages:
   The XOR operation that combines the global history with the branch PC adds computational complexity. While it enhances prediction accuracy, it comes at the cost of increased processing overhead.
   In certain scenarios where branches have very localized patterns or minimal correlation with global history, Gshare may not provide significant accuracy improvements compared to simpler predictors like local predictors.

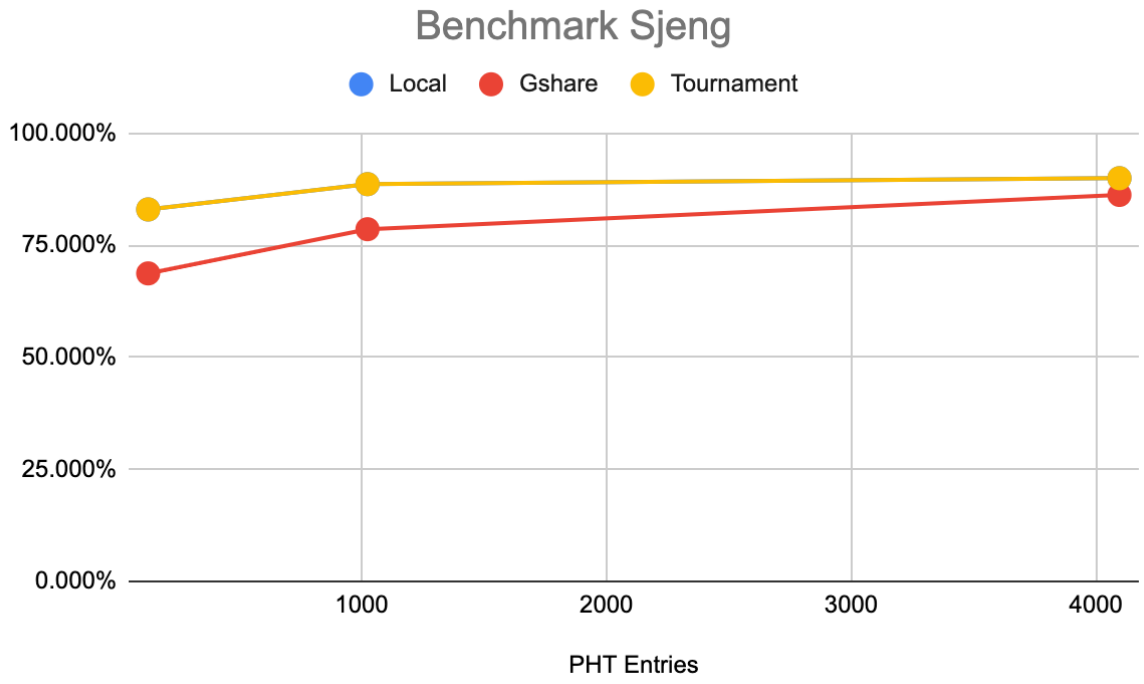
4. What benefits did you expect from the addition of the Tournament predictor? Did the results match your expectations?
   Ans: The Tournament predictor was expected to offer the advantage of adaptive prediction strategy selection. It combined the strengths of both the Local and Gshare predictors. And aims to choose the most accurate prediction approach (local or global) for a specific branch, potentially enhancing overall prediction accuracy by leveraging the strengths of both.
5. Comment on the impact of the different benchmark programs on the branch predictor accuracies. Is there a difference between the benchmarks? If so (of if not), why do you think that is the case?

Ans: The impact of different benchmark programs on branch predictor accuracies is notable. Some benchmarks may emphasize local patterns (favoring local predictor), while others emphasize global patterns (favoring gshare predictor). Matrix multiplication tends to exhibit more regular patterns, aligning with the strengths of the Local Predictor, which uses specific branch history. On the other hand, Gobmk and Sjeng demonstrate more irregular, global patterns, making the gshare Predictor more effective due to its ability to capture both local and global aspects. The benchmarks' variance in branch behavior underlines the importance of selecting an appropriate predictor tailored to the prevalent pattern in a given workload.

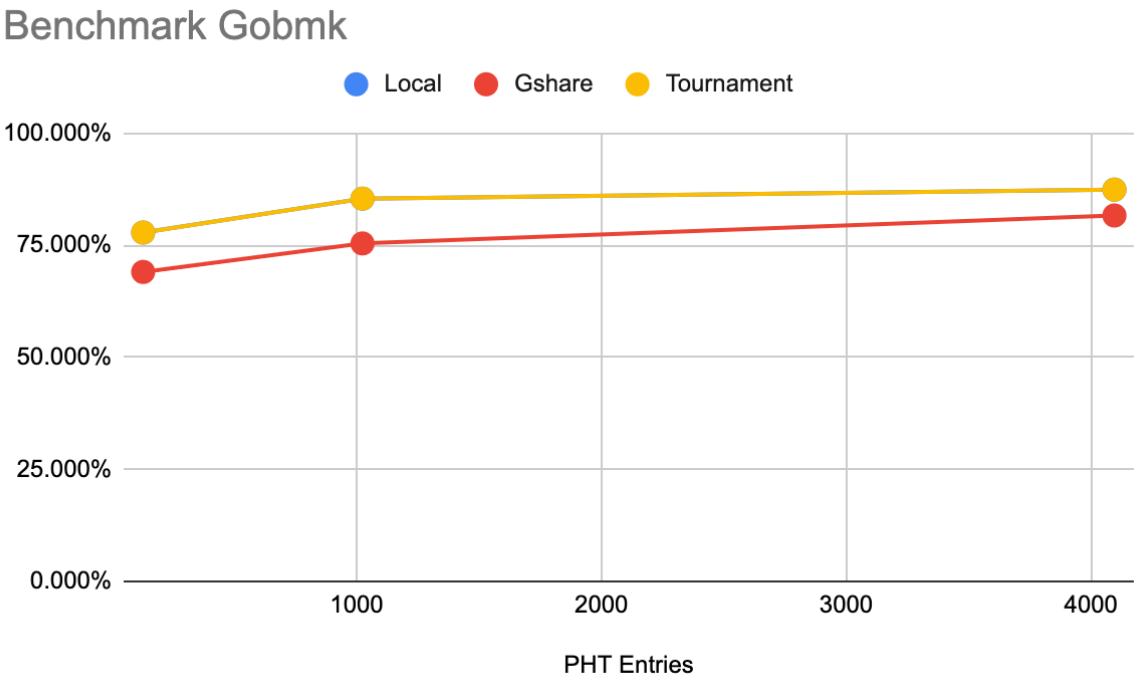
Graphs:

Local and tournament almost had the same performance here.

1. Benchmark Sjeng



2. Benchmark Gobmk



3. Benchmark Matrix multiplication