For the following code, explain:

Here is a graph of runtime vs. N, on a log-log plot.

審稿 47
Branch prediction buffers (caches)

Better:

Branch Delay slot

Statically predict not taken

Stall

Previous solutions:

Determine the right next instruction in time for instruction fetch

Branches introduce control hazards

Readings: 4.8

Dynamic Branch Prediction
Problems With Static Branch Predictors

A code: each branch has a profile/counter

if (current condition) normal case: mostly not taken

for loop i (loop test at top: branch mostly always taken

if (counter condition), exit i) branch always taken

Are all conditional branches created equal?
If incorrect prediction, annul instructions incorrectly started

Predict taken/not taken by previous execution

Direct-mapped cache w/1-bit history

Branch Prediction Buffer
Consider the following code segment:

Thought Experiment
For code on slide 74 we are 1.12% or 6.12%.
2-bit Predictor (cont.)