Goal

Through Lab 2, you would modify the branch prediction subsystem of simulator for one target: add a **perceptron-based** dynamic branch prediction algorithm that can be used by the simulated CPU core.

You can refer to the following research paper for how to make it.

D. A. Jimenez and C. Lin, *Dynamic branch prediction with perceptrons*, Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture, Monterrey, Nuevo Leon, Mexico, 2001, pp. 197-206, doi: 10.1109/HPCA.2001.903263.

You need to prove that your perceptron-based branch predictor is sound, effective and efficient. A comparison to existing branch prediction algorithm of simulator is essential.

Lab 2 will take **3 points** overall for CS211 in Fall 2022.

Due date for Lab2

23:59:59, 30th October, 2022. Any submission past the deadline shall not receive any points for this lab.

Deliverable

1. A report in English of your design, implementation, test, etc. In particular, you need to compare your branch prediction algorithm to simulator's existing branch prediction algorithms with quantitative analysis. Please record the parameters you set in the experiment report(like hardware budget in kilobytes, history length, etc.)

2. Source code for task with git commit history: the compressed file which contains source code with comments, related test programs, etc. Submitted source codes must be with comments and descriptions.

All of them shall be submitted on ShanghaiTech Cloud Driver (Campus Network Only or via VPN). The submission link is as follows.

Submission Link: http://pan.shanghaitech.edu.cn/cloudservice/outerLink/decode?c3Vnb24xNjY1NTYyNzIxNjk4c3Vnb24=

The task of Lab2

1. Read the aforementioned paper *Dynamic branch prediction with perceptrons* for reference.
2. Implement the strategy described in the paper.

   Use the below parameters as your default configuration:

   - Hardware budget in kilobytes: 8
   - Threshold θ: 79
   - History Length: 34
   - Representation of weights: signed integers

   We might use these parameters to check the correctness of your program.

3. Make sure you add at least one option for configuring perceptron-based branch prediction.

We would use "PERCEPTRON" option to test your program just like below command.

```
# An example use ackermann function to test your strategy
# And make sure this command will use the parameters mentioned in the above subsection.
./Simulator ../riscv-elf/ackermann.riscv -b PERCEPTRON
```

Note:
Although you are given a few days to do such a branch predictor, it is strongly recommended that you start this lab as early as possible.

Ref:

1. D. A. Jimenez and C. Lin, "*Dynamic branch prediction with perceptrons,*" Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture, 2001, pp. 197-206, doi: 10.1109/HPCA.2001.903263.
2. Should you have any further questions or problems, please contact TAs (Jiang Qisheng and Hu Yanpeng).