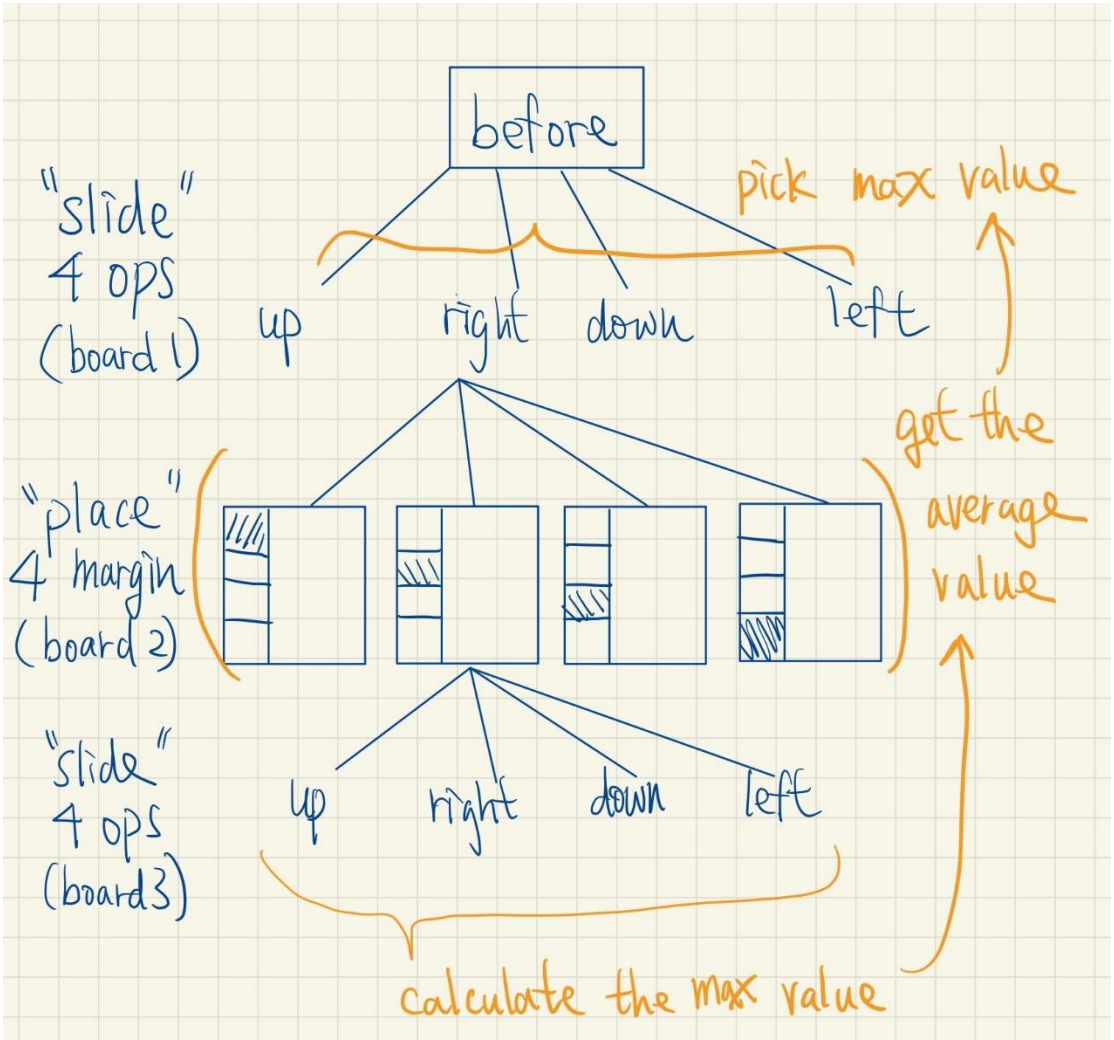
# Report of TCG Project 2+

109550027 紀竺均

a. Improvements
   1. I add expectimax search on my TD-learning model. Below is how I implement expectimax function. To sum, it expand all possible next moves and calculate next moves' expected value by place 'hint' and pretend to slide on the board.




   2. Two-step TD learning:
      Cause we have simulate "next" move and the move after next move (called it "next next" move here) from expectimax search function, I add the "next next" move's reward and its board into consideration in my TD-value function. And I use a parameter lambda to decide the value ratio between next move and "next next" move.

Two step lambda formula: (Set lambda=0.01)

```
TDerr = (1-lambda)*TDerr + lambda * (1-lambda) * ( reward + reward2 +
get_value(nextnext) - get_value(next) );
TDerr *= alpha;
```

(The reason I didn't use general TD-lambda method is that I use forward-propagation to calculate TD value, and I have no idea how to consider all the moves after current move till the end without changing my propagation model. So as simply as I can imagine, I only consider the influence of next move and the move after next move in my "Two-step TD model".)

b. Training process

Use alpha = 0.1/32 to train for 130,000 games and results in:

```
avg = 262838, max = 790629, ops = 54641 (29331|985658)
48      100%    (0.2%)
96      99.8%   (0.4%)
192     99.4%   (0.6%)
384     98.8%   (2.2%)
768     96.6%   (7.3%)
1536    89.3%   (16.2%)
3072    73.1%   (54.8%)
6144    18.3%   (18.3%)
```

Then I set alpha=0.0005 and train for another 30000 games that results in:

```
avg = 405569, max = 797664, ops = 54618 (29384|961619)
96      100%    (0.1%)
384     99.9%   (0.3%)
768     99.6%   (1.4%)
1536    98.2%   (8.5%)
3072    89.7%   (43.8%)
6144    45.9%   (45.9%)
```

Finally set alpha=0.00005 and train for another 100,000 games.

```
avg = 477461, max = 797172, ops = 56361 (30396|962268)
384     100%    (0.5%)
768     99.5%   (2.6%)
1536    96.9%   (8.3%)
3072    88.6%   (29.7%)
6144    58.9%   (58.9%)
```