Math file

## Math A

Result: Absolutely terrifying. Even a group of 1k rounds shows too many problems.

|  |  |  |  |
| --- | --- | --- | --- |
| 10000x | 1 | X100 | 1 |
| 5000x | 19 | X20 | 29 |
| 2500x | 200 | X10 | 700 |
| 1000x | 1500 | X5 | 4450 |
| 500x | 3500 | X3 | 12600 |
| 250x | 8000 | X2 | 28000 |
| 100x | 15000 |  |  |
| 50x | 20000 |  |  |
| 25x | 40000 |  |  |
| 10x | 80000 |  |  |
| 5x | 95000 |  |  |
| 2x | 220000 |  |  |
| 1x | 481000 |  |  |

## Math A2

Approach: rewrite the entire thing to make it much harder

Result: A bit too bad, Top 10 in a group of 1k only being 2175-5965.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Object | Per 10M | Total | Object | Per 10M | Total |
| 10000 | 3 | 9856600 | x100 | 2 | 143400 |
| 5000 | 47 |  | x20 | 28 |  |
| 2500 | 300 |  | x10 | 870 |  |
| 1000 | 1250 |  | x5 | 2700 |  |
| 500 | 5000 |  | x3 | 27800 |  |
| 250 | 15000 |  | x2 | 112000 |  |
| 100 | 45000 |  |  |  |  |
| 50 | 140000 |  |  |  |  |
| 25 | 400000 |  |  |  |  |
| 10 | 800000 |  |  |  |  |
| 5 | 1600000 |  |  |  |  |
| 2 | 2350000 |  |  |  |  |
| 1 | 4500000 |  |  |  |  |

## Math A2.1

Approach: A2 but with chance of getting multipliers.

Result: This time it’s 10k instead of 1k rounds.

Top one still only 5531, and #10 is only 2816.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Object | Per 10M | Total | Object | Per 10M | Total |
| 10000 | 3 | 9726600 | x100 | 2 | 273400 |
| 5000 | 47 |  | x20 | 28 |  |
| 2500 | 300 |  | x10 | 870 |  |
| 1000 | 1250 |  | x5 | 9700 |  |
| 500 | 5000 |  | x3 | 76800 |  |
| 250 | 15000 |  | x2 | 186000 |  |
| 100 | 45000 |  |  |  |  |
| 50 | 140000 |  |  |  |  |
| 25 | 400000 |  |  |  |  |
| 10 | 800000 |  |  |  |  |
| 5 | 1580000 |  |  |  |  |
| 2 | 2360000 |  |  |  |  |
| 1 | 4380000 |  |  |  |  |

## Math A2.2

Approach: I threw in more multi.

Result: in 10k rounds the highest is 8000, and a lot of top 10 rounds used 3-6 C symbols. Remember, In this version I made those C work like big bamboo or boss bear or razor ways where it collects from other C without removing them.

Also did I mention the avg result is like 250+? I just made the biggest RTP black hole ever.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Object | Per 10M | Total | Object | Per 10M | Total |
| 10000 | 3 | 9556600 | x100 | 2 | 443400 |
| 5000 | 47 |  | x20 | 28 |  |
| 2500 | 300 |  | x10 | 1270 |  |
| 1000 | 1250 |  | x5 | 12700 |  |
| 500 | 5000 |  | x3 | 96000 |  |
| 250 | 15000 |  | x2 | 333400 |  |
| 100 | 45000 |  |  |  |  |
| 50 | 140000 |  |  |  |  |
| 25 | 400000 |  |  |  |  |
| 10 | 800000 |  |  |  |  |
| 5 | 1500000 |  |  |  |  |
| 2 | 2350000 |  |  |  |  |
| 1 | 4300000 |  |  |  |  |

And I noticed this round.

1st Spin

["2", "2", "1000", "1", "2", "C", "5", "2", "1", "25", "2", "1", "5", "1", "5", "1", "25", "1", "1", "1", "1", "2", "2", "10", "2"]

Collect

["", "", "", "", "", "s1100", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""]

2nd Spin

["2", "2", "1000", "1", "2", "s1100", "C", "5", "2", "1", "25", "2", "1", "5", "1", "5", "1", "25", "1", "1", "1", "1", "2", "2", "10"]

Collect

["", "", "", "", "", "s1100", "s2198", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""]

3rd Spin

["250", "1", "1", "5", "2", "s1100", "s2198", "50", "5", "5", "1", "2", "5", "5", "5", "1", "10", "5", "5", "10", "2", "1", "5", "25", "2"]

Where the 1st spin and 2nd spin is too similar. Going to fix some bugs.

## Math A2.3

Approach: More high value coins but less multi (back to 2.5% total)

Changed 500x coin to 1/1250 (once per 50 board).

Result: avg result is now 273.24 still needs rework

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Per 10M | =% per block | Total | Object | Per 10M | =% per block | Total |
| 10000 | 3 | 0.0000% | 8301000 | x100 | 2 | 0.0000% | 250000 |
| 5000 | 47 | 0.0005% |  | x20 | 43 | 0.0004% |  |
| 2500 | 450 | 0.0045% |  | x10 | 1155 | 0.0116% |  |
| 1000 | 1750 | 0.0175% |  | x5 | 14700 | 0.1470% |  |
| 500 | 8000 | 0.0800% |  | x3 | 57100 | 0.5710% |  |
| 250 | 23750 | 0.2375% |  | x2 | 177000 | 1.7700% |  |
| 100 | 65000 | 0.6500% |  |  |  |  |  |
| 50 | 151000 | 1.5100% |  |  |  |  |  |
| 25 | 375000 | 3.7500% |  |  |  |  |  |
| 10 | 875000 | 8.7500% |  |  |  |  |  |
| 5 | 151000 | 1.5100% |  |  |  |  |  |
| 2 | 2350000 | 23.5000% |  |  |  |  |  |
| 1 | 4300000 | 43.0000% |  |  |  |  |  |

## Math A3

Added BONUS Symbol, which pays based on the amount of BONUS landed.

3 BONUS: 80x

4 BONUS: 200x

5 BONUS: 500x

Chance of getting BONUS per block is determined by the existing BONUS symbol count. Use CTRL+F to search

*bonus\_chances =*

in the code.

The odd of getting multiplier is also increased, to a total of 2.4%

Also added more fields in the database to easier to figure out what’s going on.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Per 10M | =% per block | Total | Object | Per 10M | =% per block | Total |
| 10000 | 3 | 0.0000% | 9760000 | x100 | 15 | 0.0002% | 240000 |
| 5000 | 117 | 0.0012% |  | x20 | 185 | 0.0019% |  |
| 2500 | 480 | 0.0048% |  | x10 | 1800 | 0.0180% |  |
| 1000 | 2250 | 0.0225% |  | x5 | 9500 | 0.0950% |  |
| 500 | 11000 | 0.1100% |  | x3 | 57500 | 0.5750% |  |
| 250 | 27900 | 0.2790% |  | x2 | 171000 | 1.7100% |  |
| 100 | 67000 | 0.6700% |  |  |  |  |  |
| 50 | 170000 | 1.7000% |  |  |  |  |  |
| 25 | 370000 | 3.7000% |  |  |  |  |  |
| 10 | 770000 | 7.7000% |  |  |  |  |  |
| 5 | 1350000 | 13.5000% |  |  |  |  |  |
| 2 | 2400000 | 24.0000% |  |  |  |  |  |
| 1 | 4591250 | 45.9125% |  |  |  |  |  |

Current Time: 2024-09-05 15:04:46, Progress: Round 1 finished

Current Time: 2024-09-05 15:07:07, Progress: Round 10000 finished

So the speed is 70 rounds per second. 1M round will take almost 4 hours. My fault, can’t afford a better gaming PC.

Next step is going to be optimize code to make it run faster, as well as do a 1M rounds simulation while I’m sleeping (if the program didn’t crash for whatever reason.)

## Math A4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Coin | Per 10M | =% per block | Total weight | Multi | Per 10M | =% per block | Total weight |
| 10000 | 3 | 0.0000% | 9830000 | 100 | 10 | 0.0001% | 170000 |
| 5000 | 117 | 0.0012% |  | 20 | 115 | 0.0012% |  |
| 2500 | 480 | 0.0048% |  | 10 | 1200 | 0.0120% |  |
| 1000 | 2250 | 0.0225% |  | 5 | 9225 | 0.0923% |  |
| 500 | 16000 | 0.1600% |  | 3 | 36450 | 0.3645% |  |
| 250 | 39900 | 0.3990% |  | 2 | 123000 | 1.2300% |  |
| 100 | 85000 | 0.8500% |  | **Average multi value** | | **2.451618** |  |
| 50 | 130000 | 1.3000% |  |  |  |  |  |
| 25 | 325000 | 3.2500% |  |  |  |  |  |
| 10 | 475000 | 4.7500% |  |  |  |  |  |
| 5 | 1450000 | 14.5000% |  |  |  |  |  |
| 2 | 3000000 | 30.0000% |  |  |  |  |  |
| 1 | 4306250 | 43.0625% |  |  |  |  |  |
| **Average coin value** | | **6.86381** |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Chance of getting C on every block (default value) | | **0.0135** |
| Chance of getting C decrease for every previous C | | 0.425 |
| Minimum value of Chance of getting C |  | 0.0015 |
| (Multiply those numbers by 100 for % value) |  |  |

After 50k simulated rounds it looks much better.

Since the max win is set to 250Kx I’m aiming for a price tag of 600x thus avg value should be ~580x.

--------

Average and Median

Average result: 516.58

Median result: 228

--------

Leaderboard

# 1: 223912

# 2: 195155

# 3: 128491

--------

Top percentile

Top 1% percentile: 4725

Top 2% percentile: 2947

Top 5% percentile: 1576

Top 10% percentile: 982

Top 25% percentile: 478

--------

Bottom percentile

Bottom 25% percentile: 117

Bottom 10% percentile: 72

Bottom 5% percentile: 58

Bottom 2% percentile: 49

Bottom 1% percentile: 45

--------

Thresholds

Rounds >= 500: 12010 (1 in 4.16)

Rounds >= 1000: 4887 (1 in 10.23)

Rounds >= 2500: 1280 (1 in 39.06)

Rounds >= 5000: 459 (1 in 108.93)

Rounds >= 10000: 152 (1 in 328.95)

Rounds >= 20000: 41 (1 in 1219.51)

Rounds >= 50000: 7 (1 in 7142.86)