The Math of 2048

Group 08-08

Wong K.Z., R. Wong W.J., G. Goh H.X., T. See J.W. (1I3)

August 2018

Contents

1 The Game of 2048

1.1 History

2048 is a single-player sliding tile game made by Italian web designer Gabrielle Cirulli. It was released on the internet on 9 March 2014. It has since been released as a mobile application and has become an internet sensation.

1.1.1 Example of Game

In the report, the state of the game 2048 will be displayed in a 4 by 4 matrix. This is an example: $\begin{bmatrix} 64 & 128 & 256 & 512 \\ 2 & 16 & 8 & 8 \\ & & & \\ & & & \end{bmatrix}$

1.2 Definitions

1.2.1 Square

A square is the space where the tiles are spawned or are swiped into.

1.2.2 Empty square

A square with no tile inside. In the matrix above, the bottom two rows contain a total of 8 empty squares.

1.2.3 Array

An array is any group of squares, such that for 1 ≤ n ≤ f^2(f is an arbitrary integer), an n-array of squares is any n connected squares that can fit into a f x f grid.

In any n-array, the squares are numbered S_1,S_2 ...S_n with the condition that any tiles S_k and S_{k+1} are connected for all $k < n - 1$ (aka directly next to each other). An array with n squares is a n-array.

1.2.4 Grid

The grid is the array of connected squares in the game of 2048, a 4x4 grid in this case.

1.2.5 Tile

A tile is a coloured square-shaped object that slides from square to square. Every tile has a value. In this report, tiles are represented by numbers in the matrix.

1.2.6 Value

Each tile has a number in it, this is called the value of the tile. Every value is a power of 2.

1.2.7 Merge

In the game, 2 adjacent tiles with the same value can be merged by sliding one of the tiles in the direction of the other one. When this happens, the two tiles are combined into a tile with a value equal to the sum of the values of two tiles used to form it. This process is called merging. For example, if you slide the matrix shown below towards the left

$$\begin{bmatrix} 64 & 64 & 32 & 32 \\ 32 & 16 & 8 & 8 \\ 2 & 2 & 4 & \\ & & & \end{bmatrix}$$

The tiles 64, 32, 8 and 4 will merge to form the matrix below, assuming that no new tile has spawned after the move:

$$\begin{bmatrix} 128 & 64 & & \\ 32 & 16 & 16 & \\ 4 & 4 & & \\ & & & \end{bmatrix}$$

1.2.8 Greatest tile

The tile with the greatest power of 2 inside. In the example in 1.1.1, the tile 512 is the greatest tile.

1.2.9 Score

At the start of the game, the score is 0. At any point in the game, when any two tiles are merged, the sum of the value of both tiles is added to the score, i.e. the score at any point in the game is the sum of the values of all the tiles merged in the game.

1.2.10 Path

A path is a row of tiles in decreasing value that can be swiped together to make a bigger tile. For example, $\begin{bmatrix} 64 & 128 & 256 & 512 \\ 32 & 16 & 8 & 8 \\ & & & \\ & & & \end{bmatrix}$ shows a path that will enable the player to create a 1024 tile.

1.2.11 Bend

A bend is the number of times a path changes direction. For example, in the matrix in 1.2.10, there are 2 bends in the path.

1.3 Gameplay

1.3.1 Start of the game

The game starts with 2 tiles in the 4×4 grid. Each tile can either be a 2 or a 4. It is important to note that throughout the game, there is a 90% chance a 2 will spawn and a 10% chance that a 4 will spawn.

1.3.2 Move

At every move, these 4 steps happen:
1. The player slides in a direction, either up, down, left or right.
2. All merges that can take place will take place, with merge priority given to the tiles closer to the edge
3. The tiles are aligned to the direction of the slide.
4. A new random tile, with, again, the 90% chance that it is a 2 and a 10% chance that it is a 4, will spawn in a random unoccupied square.

1.3.3 End of Game

The game ends when there are no possible valid moves. (i.e. There is no way to swipe left, right, or down) This happens when the 4 x 4 grid is completely filled up, and no two same tiles are vertically or horizontally adjacent to each other.

1.3.4 Objective

The objective of the game 2048 is to reach the tile 2048 which is 2^{11}.

2 Our Research Questions

1. What is the greatest tile achievable?
2. What is the greatest score achievable?
3. What is the optimal strategy to win this game?
4. What is the relationship, at any point in the game, between the score and the greatest tile at that point?
Extension: How do different board shapes and sizes affect the results?
The proofs of questions 1, 2, 3 and 4 will be explained in sections 3, 4, 5, and 6 respectively.

3 Greatest Tile Achievable

We will prove that the greatest tile achievable is 2^{17}, which is 131072. This will be proven in two parts.
Firstly, we will prove that the tile 2^{n+1} is achievable in an n-array. Secondly, we will prove that the tile 2^{n+2} is unachievable in an n-array.

3.1 The tile 2^{n+1} is possible to achieve in an n-array.

Inductive hypothesis: For all n-arrays, the tile 2^{n+1} is achievable.
We start with a base case: For n = 1, the greatest tile achievable is 4. As $4 = 2^{1+1}$, the base case is proven.
We assume this works for n = k-2.
This means that we are able to form a 2^{k-1} tile in a k-2 array.
For n = k-1, we would be able to get a 2^{k-1} tile in one of the squares.
We now have another k-2 array (since the first 2^{k-1} tile cannot be replaced until a 2^k tile is formed) to build another 2^{k-1} tile, which is possible.
Then, we just have to slide these two tiles together to form a 2^k tile.
This means that we will be able to form a 2^k in a k-1 array. Thus, this proves that 131072, or 2^{17}, is achievable in a 16-square (4 x 4) grid.

3.2 The tile 2^{n+2} is impossible in an n-array.

Inductive hypothesis: 2^{n+2} is impossible to achieve in a n-array. Base case: 8 is impossible to achieve in a 1-array, as the tile spawned can only be 2 or 4.
Inductive step: Let us assume that it is actually possible to achieve the tile 2^{n+2} in some n-array. For the tile 2^{n+2} to have been achieved, we need to have two 2^{n+1} tiles on the board at the move before. We must have had a point in time where there was only one 2^{n+1} tile on the array.. Thus, we can conclude that the other 2^{n+1} tile was made in a (n-1)-array. Thus if a number n fulfils the condition that a 2^{n+2} tile can be made in some n-array, the number (n-1) fulfils the same condition. Thus we can conclude by induction that the tile 2^3 can be made in a 1-array. However, by the base case, this is false. This is a contradiction, thus, the assumption is wrong and we have proven the desired result. As 2^{16+2} or 262144 is impossible and 2^{16+1} or 131072 is possible, we have proven that 131072 is the greatest tile achievable.

4 Greatest Score Possible

4.1 Premises

Firstly, using two 2s to form a 4 tile earns 4 points but using one 4 tile earns none. So, we assume that all tiles spawned are 2 as a greedy case to aim to achieve the maximum score.

4.2 Proof

4.2.1 Maxed-out game state

The state of the game that will produce the greatest score is as such:

$$\begin{bmatrix} 131072 & 65536 & 32768 & 16384 \\ 1024 & 2048 & 4096 & 8192 \\ 512 & 256 & 128 & 64 \\ 4 & 8 & 16 & 32 \end{bmatrix}$$

This is because the greatest possible tile in a 16-array is 131072. The tile 131072 cannot be removed once it has been formed, so we are left with a 15-array after forming it. Then, the greatest possible tile we can form with a 15-array is 65536, which also cannot be removed once it has been formed, so we are left with a 14-array after forming it. Continuing, we will find that the matrix above shows the game state that produces the greatest possible score.

4.2.2 Greedy calculation

The tile 131072 will be made, thus the score 131072 will be added to the score.

To make the tile 131072, two of the tile 65536 are needed. Also, another 65536 tile will occur in the end state. Three 65536 tiles are needed, and thus 65536 will be added 3 times to the score.

To make the tile 65536, two of the tile 32768 are needed. Also, another 65536 tile will occur in the end state. Seven 32768 tiles are needed, and 32768 will be added to the score 7 times.

There is a recursive relation within the number of times the value 2^n will be added to the score. Let the number of times the value $2^{(18-n)}$ will be added to the score be $f(n)$. We can conclude that

$$\begin{cases} f(1) = 1 \\ f(n-1) = 2[f(n)] + 1 \end{cases}$$

Note that in $f(n)$ is equal to $2^n - 1$, as $f(1) = 2^1 - 1$, then $f(n-1) = 2[f(n)] + 1 = 2(2^n - 1) + 1 = 2^{n+1} - 2 + 1 = 2^{n+1} - 1$

Thus the greatest possible score, so far, in this greedy calculation is

$$\sum_{i=1}^{16} (2^{18-i})(2^i - 1) = 3933188$$

4.2.3 3933188 is not the greatest possible score

Why? There are sixteen 4s necessary in achieving the maxed-out game state, though. Look at the configurations. In all these, a tile valued 4 must spawn at the space marked x.

$$\begin{bmatrix} 65536 & 32768 & 16384 & 8192 \\ 512 & 1024 & 2048 & 4096 \\ 256 & 128 & 64 & 32 \\ x & 4 & 8 & 16 \end{bmatrix}$$

$$\begin{bmatrix} 131072 & 32768 & 16384 & 8192 \\ 512 & 1024 & 2048 & 4096 \\ 256 & 128 & 64 & 32 \\ x & 4 & 8 & 16 \end{bmatrix}$$

$\vdots$

$$\begin{bmatrix} 131072 & 65536 & 32768 & 16384 \\ 1024 & 2048 & 4096 & 8192 \\ 512 & 256 & 128 & 64 \\ x & 4 & 16 & 32 \end{bmatrix}$$

Note that every time before one of the tiles in the maxed-out game state is made, we have to spawn a 4 tile.

4.2.4 Final maximum score

By the premise, we have to minus 64 from our greedy case as we spawned 4 sixteen times. Thus the greatest score is

$$(\sum_{i=1}^{16}(2^{18-i})(2^i - 1)) - (16 \times 4) = 3933124$$

.

5 Optimal strategy

The best strategy in the game is the strategy that will give the highest chance of reaching the tile 2048 in the game. As such, we have found that the best strategy that we have found is the snake-line strategy. We will prove that this is the best strategy in the game.

5.1 Connections

In order to get a 2048 tile, we need two 1024 tiles. In turn, to get a 1024 tile, we need two 512 tiles. Essentially, we need two tiles of value n to be connected in order to make a tile of value 2n. Using this, we found that the best way to connect tiles is using a method that will yield in the least number of bends.

5.1.1 Method with least bends

The least number of bends needed to arrange n^2 tiles in an n-array is, by 2(n-1) bends. This can be proven by construction, using a set up as follows: starting from the top-left corner of the grid, we arrange the tiles from left to right until we hit an edge, thereby forcing us to turn to an up-down arrangement, before we hit the bottom edge, and start to move from right to left... etc, Here is an example where we number the first tile 1 and the last tile in the centre 16. Asterisks mark where we need to make a bend.

$$\begin{bmatrix} 1 & 2 & 3 & 4* \\ 12* & 13 & 14* & 5 \\ 11 & 16 & 15* & 6 \\ 10* & 9 & 8 & 7* \end{bmatrix}$$

This results in a spiral-shaped path where we only make bends when needed. As seen, we need 6 bends, which is the least. The snake line also uses 6 bends (see below) and is much easier to achieve as we only need to swipe in three directions to align them into the snake line path. This leaves less room for error. Thus, the snake line is the best method to use.

$$\begin{bmatrix} 1 & 2 & 3 & 4* \\ 8* & 7 & 6 & 5* \\ 9* & 10 & 11 & 12* \\ 16 & 15 & 14 & 13* \end{bmatrix}$$

5.2 Why go for the least bends

The lesser the number of connections, the lesser the chance of error. This is because when you slide to make a connection along a bend, a tile will unavoidably move into the wrong place such that your connections will be destroyed.

$$\begin{bmatrix} 32 & 128 & 256 & 512 \\ 8 & 32 & 16 & 4 \\ & & & 2 \\ & & & 2 \end{bmatrix}$$

In this scenario, which arises from the snake line strategy, our path is disrupted around the bend. This is almost inevitable, as to form the tiles 512, 256 and 128, the tiles have to be flushed to the right as any move to the left could result in the 512 tile moving out of position. However, to make the 32 tile in the space at the top left hand corner become a 64 tile, tiles have to be flushed to the left as the square which the tile is in is at the far left and it is safe to do so as the top row is rendered immobile.

To continue in our quest of making the tile in the square in the top left corner a 64, we make the square below it a 32. However, you may get something like this:

$$\begin{bmatrix} 32 & 128 & 256 & 512 \\ 32 & 32 & 16 & 4 \\ 4 & 4 & & 2 \\ & & 4 & \end{bmatrix}$$

After sliding up, (any other move will bear the less optimal outcome of breaking the path), we are still not in the formation we need to be in. Thus we must make the tile in the square below the 64 take on the value 64 (see matrix below), only then we will get into formation again.

$$\begin{bmatrix} 64 & 128 & 256 & 512 \\ 4 & 32 & 16 & 4 \\ & 4 & 4 & 2 \\ & 2 & & \end{bmatrix}$$

For every bend we meet, we could land into this common scenario and move progressively further from our desired formation. Every time we break out of formation, we have an increased risk of taking up more squares than we need to, increasing the risk of occupying all 16 tiles and losing the game. This is why we must minimise bends.

5.3 Problems with the best method

When you slide to make a connection along a bend, a tile might inevitably spawn in the wrong place such that one will be forced to slide in a direction that will break the path. This happens in all strategies, and is especially prevalent at bends.

$$\begin{bmatrix} 64 & 128 & 256 & 512 \\ & 16 & 8 & 4 \\ & & & 4 \\ & & & \end{bmatrix}$$

If the player decides to slide the 4 up from the current position, since a new move will spawn a new tile, a tile might spawn in the square next to the 16.

$$\begin{bmatrix} 64 & 128 & 256 & 512 \\ 2 & 16 & 8 & 8 \\ & & & \\ & & & \end{bmatrix}$$

Thus, the path is now broken as the player is stuck with a 2 at the left side. If the player, again, unknowingly makes a series of bad moves such that he is stuck with this scenario:

$$\begin{bmatrix} 64 & 128 & 256 & 512 \\ 2 & 32 & 8 & 4 \\ & & & \\ & & & \end{bmatrix}$$

The player is now left with only 1 legal move which is to swipe down.

$$\begin{bmatrix} & & & \\ & & & \\ 64 & 128 & 256 & 512 \\ 2 & 32 & 8 & 4 \end{bmatrix}$$

After that, a tile will spawn in the top 2 rows.

$$\begin{bmatrix} & 4 & & \\ & & & \\ 64 & 128 & 256 & 512 \\ 2 & 32 & 8 & 4 \end{bmatrix}$$

When the player tries to salvage the situation by swiping up again, he will be stuck with again, a broken path.

$$\begin{bmatrix} 64 & 4 & 256 & 512 \\ 2 & 128 & 8 & 4 \\ & 32 & & \\ & & & \end{bmatrix}$$

This results in a quagmire for the player. It is very difficult to reverse the damage done.

5.4 Conclusion

As shown, for every bend we meet, we could land into the common scenario shown above, where the path breaks out of formation. This results in a game that is very difficult to win at.

It is also shown that when the path breaks out of formation, we have an increased risk of taking up more squares than we need to, thus increasing the risk of occupying all 16 tiles and losing the game.

6 Relationship between the score and the greatest tile possible at that point in time, in any point of the game.

There is a lower bound for the score if the largest tile on the 4×4 board is 2^n.

6.1 Lower bound

Let $f(n)$ be the recursively defined function to find the lower bound for the tile 2^n. Then, we shall prove that $f(n) = 2[f(n-1)] + 2^n$, where $f(3) = 8$.

6.1.1 Premise

Using two 2s to form a 4 tile earns 4 points but using one 4 tile earns none. So, we assume that all tile spawns are 4 as a greedy case to achieve the lower bound.

6.1.2 For $n = 3$

If $n = 3$, the largest tile is 8, and a minimum score of 8 can be achieved (using 2 4s to form a 8).

6.1.3 Proof of function

Let $f(n-1)$ be the lower bound of the score needed to obtain a 2^{n-1} tile. Then, $f(n)=$ (the sum needed to make 2 2^{n-1} tiles) + (the score obtained from combining these two tiles together to form a 2^n tile), which makes $f(n) = 2[f(n-1)] + 2^n$. This result can be also be factorised into $f(n) = 2[f(n-1) + 2^{n-1}]$.

6.2 Upper bound

6.2.1 Problems

This question proves too challenging to solve as it seems that only brute force and construction strategies work. The lemma we have found is flawed. Nonetheless, even with the lemma, we are unable to prove whether such scenarios are possible, which can only be proven, as far as we know, by construction of individual cases. It is near impossible to generate all the possible game states due to the random generation of numbers in-game, and the resultant size of the game tree. Therefore, we cannot be certain that what we have shown in our lemma is the true upper bound. As such, this question remains unsolved.

7 Conclusion

Out of the four research questions, three have been solved. It has been proven that the greatest tile achievable in an n-array is 2^{n+1}, thus in the game of 2048, the greatest tile achievable is 2^{17}. For the second research question, our findings show that the greatest score possible is 3933124. As for the third research question, our proof shows that the optimal strategy to win the game is the snake-line strategy where bends are minimised. However, the fourth research question requires further investigation, due to the complexity of the problem. For further research into 2048, we would like to investigate the game with different array shapes and sizes.

8 References

What is the largest tile possible in 2048? (n.d.). Retrieved April 02, 2018, from https://puzzling.stackexchange.com/questions/48/what-is-the-largest-tile-possible-in-2048

Hoang, L. N. (2016, January 29). The Addictive Mathematics of the 2048 Tile Game. Retrieved April 02, 2018, from https://www.science4all.org/article/2048-game/

Singh, K. (2015, December 21). Artificial Intelligence for 2048. Retrieved April 02, 2018, from http://iamkush.me/an-artificial-intelligence-for-the-2048-game/