The judges scoring, note	Team number： 20035	The judges scoring, note
The judges scoring, note	Problem： A	The judges scoring, note

2048 gameplay

Abstract

After the game “FlappyBird”off the shelf,,a new game ,named”2048”,became very popular among the people.Many netizens said it looks easy to play,but actually not.In order to solve the problems in relation to this diffficult game,we found the following model.

In terms of question 1， Starting from the essence of combination of recursive manner ,we got the basic steps of “2048”,and then , we combine the mini-max Algorithm principle and Play Steps ,we got the generic model.When calculating the times of box movement , considering the Randomness of last figure’s appearance,we adopt the method of Backward and finished the steps of “2048”.Finally,according to Alpha—beta Pruning algorithm,we gathered and analyzed the statistics of random player,found a new math model in which player reached the Maximum number of steps and success rate.Through the Comparative analysis between new model and generic model,we found the latter is effective.

In terms of question two,Using Recursive algorithm and Mathematical induction, we got the Max-model .Then , we discussed the maximum number of “Q”values when M in different values .Finally,we got the maximum number of “Q”values is .Suppose we can get the maximum is ,after calculating, the hypothesis wasn’t set up,so the maximum “2048”can reach is .Using the Max-model,we found the Recursive algorithm .Suppose the maximum of the model is ,Then, through calculating,we found the hypothesis is right, so , the maximum of the game is

Keyword： *Mini-max* algorithm *Alpha—beta* pruning algorithm recursive algorithm

"2048" Gameplay

1. Problem restatement

"2048" is a puzzle game，Its rules of the game: Each box controls all movement in the same direction, the same two-digit block collided and merged to become, Blank squares in a randomly generated number 2 or 4 after each operation,finally a box get "2048" will clearance. If the digital grid within 16 to fill the entire grid and the adjacent not the same, you can not move the grid, the end of the final game.

Through the establishment of the model to solve the following problem:

1. How to reach 2048, given a generic model, and to verify the validity of the model used to complete the game to move the required number and probability of success;

2.btained after 2048, can continue to play the game, then the maximum attainable value is how much? If extended to N × N square, the largest number is the number that can be achieved?

2. Analysis

For questions of a lot of friends call themselves " Once it simply can not stop playing on down," Based on this , the number of users by gathering test , calculated that 2048 free play is difficult to achieve , and therefore need to establish a common model . Because when free to move boxes , if the combination of two of the same number the greater the number , the harder grouped together , found through experiments , the basic idea of this game is recursive generation , so hold the greatest value , and then accumulated through recursive the method to obtain the new maximum number. According to this model the idea by mini-max algorithm, and given to the model , it is assumed the mobile number . Among verify the validity of the model , play the game randomly selected 100 people to test , the success rate is calculated , therefore , concluded by several experiments . When mobile number and verify the effectiveness of the model , the number of moves to get established at the end of each game with the maximum number of scatter plot , the regression equation , when a combination of a number of mobile number 2048 if needed , is similar to the assumptions, such as assuming similar verify the validity of the model.

For question two , according to the recursive generated models , according to the general problem of a given model , the maximum number in one corner , and the rest into a snake once decreasing grid penultimate number is 4 and the last number generated by the system 4:00 , squares before they can move , the game continues, the cumulative recurrence obtained by the last digit , while the remaining 15 squares method in accordance with the recursive accumulation , the last one to get the maximum number of digits that easy the first number was calculated as the maximum , so get the " 2048 " the greatest combination of value , we can draw this endless game play can not go on . For the grid by mathematical induction , summed square law model , the conclusion of the problem.

3. Model assumes

1Assuming all valid mobile every move;
2 Assuming each combination can only be a combination of a pair;
3 Assuming random number system regardless of the location of the test results;
4 Assuming the game has nothing to do with the number of steps and the player's own subjective factors.

4. Symbol Description

	Finally, a random number out of the system
	The maximum number of grid generation
	The maximum number of combinations of the end
	When the end of the maximum number of digits
	Mobile number
	Number of steps that appear to move
G	All games in a finite set of models that can be achieved under ideal
k	Successor function
m	Gameplay Stats
	Instead of a problem in 2048
	number of successes
	The maximum number obtained at the end of each game
	Reached the number of maximum
	The average number of steps
	Number of power
	Plaid number

5. Establish and solve the model

5.1 issues a

5.1.1 Background

"2048" is the latest launch of a game, the rules of the game are as follows:

- Random two numbers in the box • When you start, just may be 2 or 4;
- Players can choose four directions, if the digital board inside the displacement or consolidation, as valid move;
- If the player chooses the same direction of the figures combined, each valid move simultaneously merge, but not continuous consolidation;
- Merge the resulting figures would like to add all the new generation of the step is valid score;
- The direction the player chooses a row or column appears in front of a space displacement;
- Effectively move every step, the board space (no figures at) random digit (still may be 2 or 4);
- The board is filled with figures, unable to move effectively, forfeit, the game ends;appears on the chessboard 2048, sentenced to win.

By collecting data[1] , the users played a total of 4246 times the maximum number appears (as shown in Table 1) and the number of moving boxes that appear at the end (see Table 2) the end of each:

Table 1 the maximum number of occurrences

	1	7	218	1412	2173	433	2

Table 2 times the number of moving boxes and appear

Y	0~50	50~100	100~150	150~200	200~250
	2	362	1458	1556	551
Y	250~300	300~350	350~400	400~450	450~500
	247	64	4	1	1

Based on the collected data obtained, a minimum value in Table 8, the maximum value can reach 1024, 101 calculated mathematical expectation; minimum value in Table 2 is 41, the maximum value is 468, the median value 161, 163 is the mathematical expectation .

Thus, random play is more difficult to reach 2048, but with a multi-step, to establish the following general play, increasing the success rate reached 2048 and reduce the number of moves.

5.1.2 Mini-max algorithm model

Mini-max algorithm (Figure 1) is a regression algorithm, which means that the search performed after the game tree to the leaf node, to obtain the value of the ending, and then pass these values to the process according to the Mini-max up until Get all internal nodes Mini-max value [2].

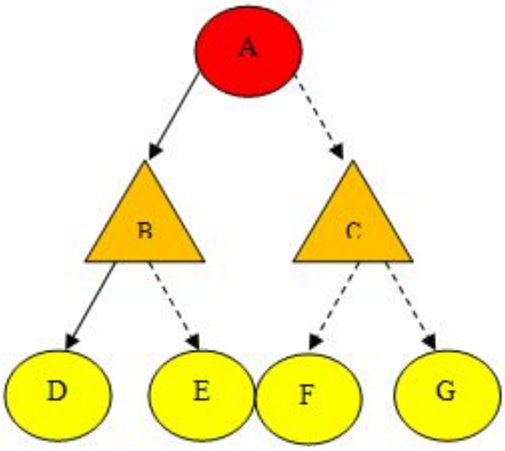


Figure 1 Mini-max tree algorithm

Up to 2048 for the establishment of a common model, defined as follows:

An interval assigned to a game quaternion equations (*G, m, k, f*), where *G* is a finite set of all games can be achieved in an ideal model, the initial state of the game, k is the successor function, the evaluation function is .

Said state by state m after a reasonable step is completed, the game is usually established model should meet the following two conditions:

(1) state cycle back and forth does not exist.

(2) At the end of the game.

Among them, 0 and 1, respectively, on behalf of those who run the game after the completion of the end of the checkpoints and finish continues to go on, the game outcome is 0, become minimizer, the outcome is 1 called maximizer, on behalf of the middle value between 0 and 1 As a result, a situation that is able to fulfill the objectives. After the completion of the case on behalf of checkpoints continue to go on, after the end of the case on behalf of the completion. Recursively defined functions:,as follows:

From the above equation, the value max is equal to the results of the maximum value, min is equal to the minimum value of the results, the mini-max obtain the following algorithm:

```
If    return
Else   if      return
Else   if        return
```

5.1.3 Alpha-beta pruning algorithm

Single *Mini-max* algorithm to obtain a generic model is complex, due to the large number of digits to be constructed, the steps need more and therefore limits the number of its movement.

The use of *Alpha-beta* pruning algorithm works (Figure 2), from the beginning of the greatest figures detailing each step of using *Alpha-beta*:

(1) is replaced by the maximum number, the minimum number is two.

(2) the number of nodes to draw .

(3) a third layer structure node, ibid.

(4) until a configuration of up to 2 nodes.

(5) The remaining structure of the four steps described above is completed.

For question one, according to *Alpha-beta* pruning algorithm steps to draw schematic diagram of 2048. That thought synthesis, first consider where the tree 1024, to form 1024, first synthesized two 512 and adjacent to ensure that these two figures. Similarly, each of the synthesis requires two adjacent 512 256, and so on, until the minimum number of two branches to date (Figure 2), which can quickly calculate the number of steps required.

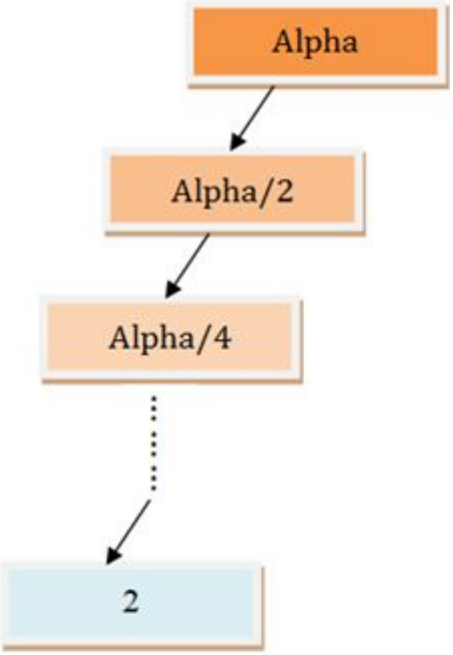


Figure 2 *Alpha-beta* pruning algorithm schematic

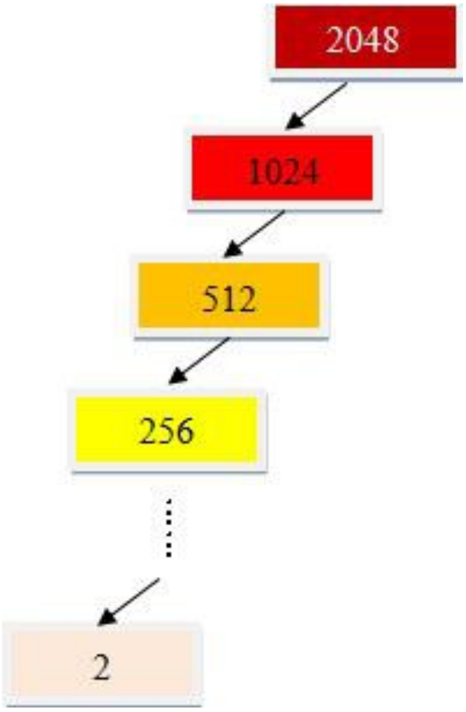
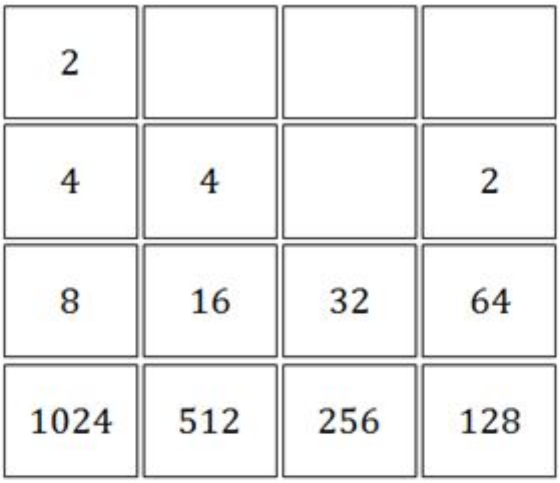


Figure 3   2048 figure is derived

When the figure is constructed by digital size in order (from high to low) will 1024,512,256,128 four rows of the edge, the four number does not move, when the digital increasingly large, stacked to the corner of the big figures in the past, so you can try to reduce grid usage. Note: If the figures given in the lower right corner, only to be moved down to the right or to ensure that large numbers do not move, so the first thing to do is to line up where large numbers of a row or column, the four fill the grid, arranged in descending order is still out. When a row or a column to fill up, you can get to the left, down, right. If the numbers are more and more needs to be done is to figure serpentine arrangement, the first line of butt ending in the second line of the first element, so as to ensure the most economical grid, but also the most secure digital stacked high. Figure 4 is a case of perfectly aligned, perfect merge into the last step 2048.



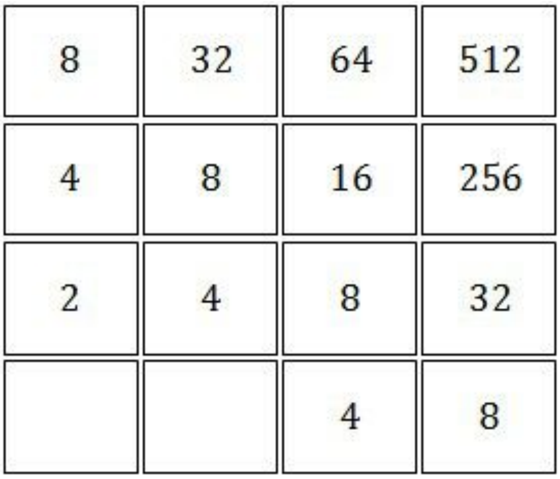
2			
4	4		2
8	16	32	64
1024	512	256	128

Figure 4 Synthesis 2048

Pattern evaluation: high scores, "good" pattern is the key to success;

Conversely, low scores, because "bad" pattern caused.

Monotonicity: Monotonicity refers to the number of the box from top to bottom, from left to right are incremented or decremented to follow. In general, the better the pattern of the monotone, as shown in Figure 5



8	32	64	512
4	8	16	256
2	4	8	32
		4	8

Figure 5 monotonous pattern

Number of spaces: The less space the more unfavorable for the player, because the less mobile digital space, the less chance for players to choose.

5.1.3 box model mobile number

The end of the game when the advent of digital 2048, are assumed to be random 2, and each can and to synthesize and only a few make it in turn generates:

- The first step you need to move a generation four times; (Compositiod by double 2)
- Second generation 8 needs to be moved twice; (adding another 4 respectively by the two 2)
- The third step is to generate 16 to be moved four times; (adding another eight were from the four 2)
- The fourth step needs to generate 32 to move eight times; (another 16 are added by 8 2 respectively)
- The fifth step to generate 64 to be moved 16 times; (adding another 32 by the 16 2, respectively)
- The sixth step to generate 128 to be moved 32 times; (adding another 64 by the 32 2, respectively)
- The seventh step of generating 256 to be moved 64 times; (adding another 128 by the 64 2, respectively)
- The eighth step needs to be moved to generate 512 128; (another 256 are added by a 2 respectively)
- Step ninth generation 1024 need to move 256; (another sum of 256 2 512 respectively)
- Step tenth generation 2048 needs to be moved 512 times; (another sum of 512 2 1024 respectively)

Therefore, the need to move the total number of:

When the figures are random digital 4:00, and each can and into the synthesis and only a few make it in order to generate:

- The first step to generate 8 requires a step; (2 4 Composition)
- The second step requires two steps to generate 16; (another 8 consisting of two 4)
- The third step is to generate 32 requires four steps; (another 16 consisting of four 4)
- The fourth step requires 64 steps to generate 8; (another 32 consisting of eight 4)
- The fifth step to generate 128 requires 16 steps; (another 64 consisting of 16 4)
- The sixth step of generating 256 requires 32 steps; (another 128 composed of 32 4)
- The step seventh generation 512 requires 64 steps; (another 256 composed of 64 4)
- The step eighth generation 1024 requires 128 steps; (another 512 consisting of 128 4)
- The step ninth generation 2048 256 steps; (another 1048 consisting of 256 4)

Therefore, the need to move the total number of:

When 2 or 4 are present , can not determine the probability of appearing . However, after a search of the game source code (see Appendix [1]) , there can be seen two of the probability of 90%, there is a 10% probability of four .

- Generate 2048 need to move 1 (2 1024 together)
- Produce two 1024 need to move two times (4 512 respectively adding)
- The need to move to produce 4 512 4 (8 256 respectively adding)
- Produce 8 256 need to move eight times (16 128 respectively adding)
- Generating 16 128 need to move 16 times (32 64 respectively adding)
- Generate 32 64 need to move 32 times (64 32 respectively adding)
- Generate 64 32 need to move 64 times (128 adding 16 respectively)
- The need to move to produce 128 16 128 (8 256 respectively adding)
- The need to move to produce 256 256 8 (512 4 respectively adding)
- Generate 512 4 461 times need to move (922 2 and 51 , respectively, adding 4)

Therefore , the need to move the total number of :

(Times)

The big game of the model space . For example , suppose that appear at each step is 4 , then at least 511 steps need to appear 2048. Each step has an average of at least two options . So that the number exceeds the total available energy . Similar to chess, Go , basically impossible algorithm can guarantee 100% win. However, if used the right methods and strategies , you will get a higher win rate.

5.1.4 Model Checking :

Success rate test:

With the general model of play the game , guess theoretically follow this approach a success rate of 100% , in order to verify the validity of the model and the success rate is close to 100% , in 2048 the population does not reach 100 randomly selected people, each play 1, you will get 100 data, and the maximum number of records at the end of the game every time they get (Figure 6, Appendix [2]) .

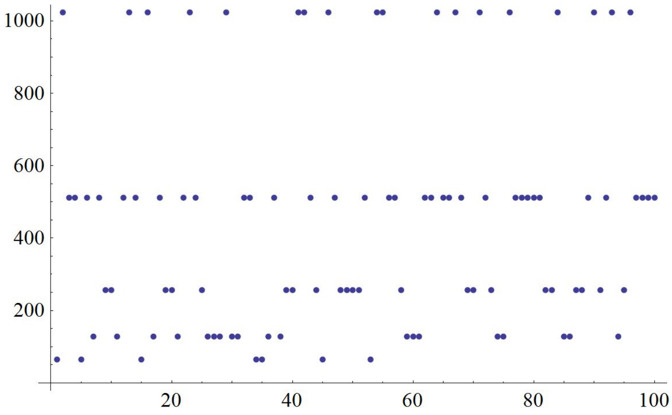


Figure 6 does not reach the 2048 record

As shown in Figure 6, the test is not found in the population rule, free to move, the maximum number is reached when the game is dispersed, and did not reach 2048. To validate the model, so that the model 100 according to the personal data given then once obtained, shown in Figure 7:

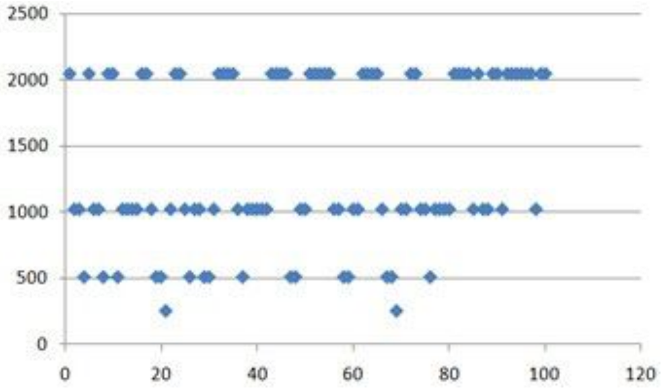


Figure 7 Test record time

As can be seen from Figure 7, the maximum number is 2048, the minimum number is 256, the mode 2048, and secondly, there is a maximum number of 1024, a success rate of 43%, though the number has reached 2048 the number of people other than the number and the success rate has improved, but the effect is not obvious, analysis of the reasons is the first test of the model, the model test population is not familiar with, but you can see the results have been better than before the trial model. In order to verify the validity of model building, on a random sample of 100 individuals in the second, third test (Figure 8 Figure 9)

Figure 8 second test records

Figure 9 Third Test Record

Figure 8, it can be seen in Figure 9, after the two scores significantly higher than the first, the third test 95% success rate, but there are some errors due to the emergence of new box fixed position, and the human also makes with the direction of movement of uncertainty, but can be seen in several training, more about model, the easier to form a fixed movement pattern, the higher success rate.

Box testing mobile number:

To verify the effectiveness of the model, randomly selected 100 players play "2048" game, playing five times for each person, they can achieve the greatest record numbers, and the average number of steps used, the results shown in Table 3:

Table 3 Maximum number of arriving and the average number of steps

64	128	256	512	1024
89	124	289	358	673

Use mathematica find the regression equation and draw a scatter plot. Get maximum figures and the average number of steps used in the regression equation:

Scatter is a model (Appendix [3]):

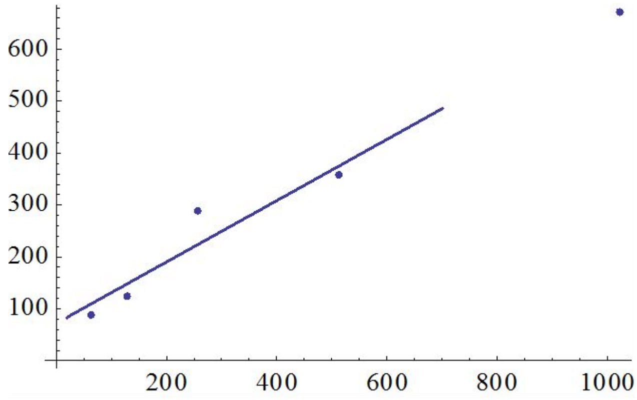


Figure 10 the number of steps and the maximum number of return

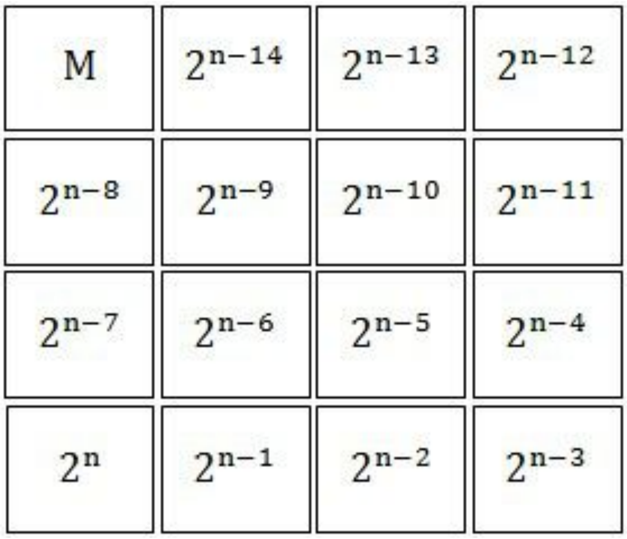
When the number reaches 2048, into the regression equation, the number of steps used for the 1281, and the establishment of a common model when there are 2, the number of steps used in 1023 (4 or 2,4 appear appear when used in step number less) than the experimental data obtained 1281. This is due to the game's own factors, such as the number of steps will lead to ineffective experimental data is too large.

To sum up: The establishment of a common model by model validation prior to reaching the maximum number of relationships and the average number of steps with established and effective.

5.2 ISSUES

5.2.1 recursively generate the maximum

"2048" The basic idea of the game is to recursively generate the largest number in the lower right corner, according to the largest number of molding in order of priority. In Figure 11, the maximum number, i.e., "2048" can reach the maximum number of games.



M	2^{n-14}	2^{n-13}	2^{n-12}
2^{n-8}	2^{n-9}	2^{n-10}	2^{n-11}
2^{n-7}	2^{n-6}	2^{n-5}	2^{n-4}
2^{n}	2^{n-1}	2^{n-2}	2^{n-3}

Figure 11 Recursive generation 2048

Since the last few M randomly generated, Points of discussion:

1 .When 时, stop the game; when the same box can be combined, the game continues, that all boxes can be followed in accordance with a combination of snake,

the final cumulative recurrence we have:

is two final composition obtained , so that the maximum number of combinations Q is :

（1）

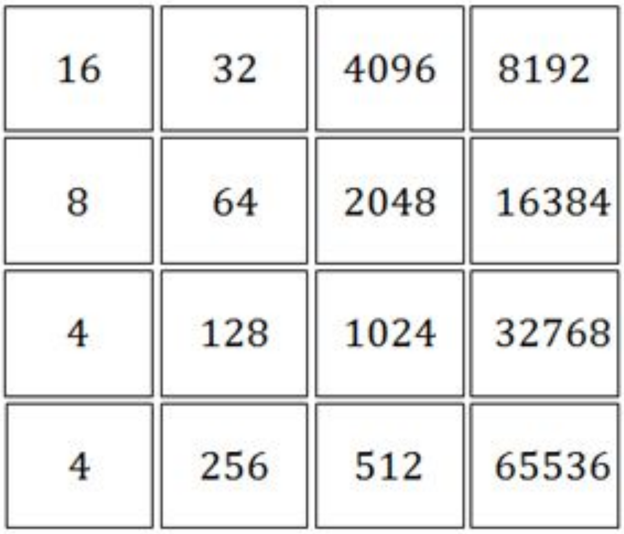
From (1) shows this case, i.e. , at the lower right corner to , the remaining 15 is the maximum number of grid . When the is taken and M 4 4 can continue the game, when may not be combined, so that the maximum number can reach .

2. When , and the game is over, the game continues, the maximum number of combinations still is at this time:

（2）

From (2) shows that this point, i.e., when the maximum number of achievable.

3.Suppose can also be synthesized, then the previous step must be synthesized by two , and assuming one of have been synthesized, then another the previous step must be synthesized by two . Suppose one of the synthesized , Previous certain other two Synthesis ...... and so on, but because "2048" is a squares, namely availability only 16. And so on, the penultimate grid space becomes , and the maximum number M of the last four, but this has no extra space to re-generate four, so no two lobbied a synthesis of to 3, so there is no two synthetic ...... and so on, so no two generating 2 to . However, if there is a maximum of to vacancies regenerated, so 2 ^ 3 ..... lobbied two and so on, it is possible the synthesis of , which can not be synthesized. To achieve the maximum value of (Figure 12)



16	32	4096	8192
8	64	2048	16384
4	128	1024	32768
4	256	512	65536

Figure 12 Is generated recursively 131072

In summary, the "2048" is not endless game play down, under the circumstances, the "2048" Analog into 11 specifications, which must be known to the upper left corner of the grid 4, in order to ensure the maximum value obtained , from the chart we can know , in this case the whole can be combined, and therefore can be combined to obtain the maximum, the number (2 or 4) and there occurs a random position, the model is provided to achieve the ideal maximum.

5.2.2 induction hypothesis

When extended to the grid squares, in order to calculate the maximum value that can be achieved using the analysis assumes that mathematical ideas.

When the number of lattice too, the random number is 4 or 2, to achieve the maximum occurs 4, the maximum value is, as shown in Table 4:

Table 4

When the grid is too few, upper left corner 2:00 unable to move, the game terminated;

When 4:00, continues to move, as shown in Table 5:

Table 5

That is the maximum achievable.

When the upper left corner of the grid reaches 2 can not move, the game terminated; When 4, the game continues, as shown in Table 6:2:00 left corner appears unable to move, the game terminated; When 4:00, the game continues, as shown in Table 6:

Table 6

Therefore, the maximum is reached.

When the grid reached, Has its maximum value is obtained that can occur.

So assuming that when the grid reached, as shown in

Table 7 Chart

Assuming can be synthesized, then it must be a two-step synthesis, and assuming one of has been synthesized, the other on the two step must be Synthesis of . Suppose one of have been synthesized, and the other on the step must ...... and so on, assuming that the squares, namely the availability of , then the upper left corner of the second number is so there is space to form 2, 2 can be generated to generate , and so on, it is possible to generate , that assumption is valid.

6. Model Evaluation

6.1 Advantages model

1 recursive thinking will only process data concluded that a clear structure , simple operation, readable, and easy to use mathematical induction to prove the correctness of the algorithm ;

(2) the model established a strong theoretical foundation of mathematics , increasing the credibility of the conclusions ;

3 ruled out once again merge multiple numbers and figures can not be combined objective factors ;

4 responses objectively authenticity and feasibility game results , to prevent subjective bias ;

According to the established model of relatively small steps with the 2048 target figures , clear, strong and regular use of this model can be composed of a greater number.

6.2 Disadvantages model

1 In practice , the game changing circumstances , but we only consider several special cases , the actual results to error ;

2 can not be the result of all the factors that affect the game are included in the calculation , so that bias .

6.3 Model Promotion

To solve this problem , using a recursive algorithm and mini-max algorithm, by modeling the data processing conclusion.

The model can be extended to the study of nine rings , chess and other issues , such as the use of the model established in this paper to solve the problem of nine serial solution , using a recursive algorithm can draw the number of steps needed to put on a chain as follows:

6.4 Model Optimization

When this model can merge multiple rule out a number of factors and ineffective merger concluded there is a deviation. Therefore, the program should take into account these two factors, and make the appropriate model, taking into account also the position of each digit appears.

References

[1] http://www.guokr.com/post/577816/

[2] Liang Bin , Hybrid Mini-max search algorithm , Zhongshan University, a master's degree thesis probabilities , 2006.

[3] Yang Aimin , Liu Chunfeng , music, Qujing Guo , university mathematics test, Beijing : Science Press , 2012.

[4] Jiang Qiyuan , a mathematical model (second edition) , Beijing : Higher Education Press , 1993.

[5] Cai lock chapter , mathematical modeling principles and methods , Beijing :

Ocean Press , 2000.

[6] Richard O.Duda, Peter E.Hart, David G.Stork, pattern classification, Machinery Industry Press , CITIC Publishing House, 2011 .

[7] Huan-Wen Tang , Heming Feng , mathematical models Introduction (second edition) , Beijing : Higher Education Press , 2002.

[8] Sun Bowen , fractal algorithms and program design and Visual C + + implementation, Beijing : Science Press , 2004.

[9] Jiang Xin , Yan Mu Jiang , Chess entry through January , Shanghai Culture Publishing House , 2005,15 .

[10] Sun Wei , Ma Shaohan Distributed game tree search algorithm , Computer Journal, Vol 18, No. 1 .

[11] Xiaoqi Ying , Wang Zhi , game tree search function with a static estimate , Computer Applications Research, 4 1997,74-76 .

[12] HJ Berliner, Some Necessary Conditions for a Master Chess Program, Procs.3rd Int.Joint Conf.on Art.Intell,. (Menlo Park: SRI), Stanfoet ,1993,77-85.

[13] Wang Xiaodong , a computer algorithm design and analysis, Beijing : Electronic Industry Press , 2001.

[14] Wang winding away , Liu Yunxia , recursive simulation process , Zhoukou Teachers college , 2000,17 (5) : 78 ~ 80.

[15] Wang Yongyan , healthy , an intuitive recursive simulation method , Jinan University, 2001,15 (3) : 240 - 242.

[16] QI Zhi-Chang , Tan Qingping , NING , software engineering (second edition) , Beijing : Higher Education Press, 2004:231 - 340.

Appendix

Appendix [1]

```
/*2048*/
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<time.h>

int code[4][4]={0,0,0,0,
                        0,0,0,0,
                        0,0,0,0,
                        0,0,0,0};/* Game 16 grid */
int temp[5];/* Intermediate variables */
int move=0;/* Mobile number */
int score=0;/* Fraction */

void print(void)/* Show the game interface */
{
    int i,j;

    clrscr();/*Clear the screen */
    printf("2048\n");
    printf("W--UP    A--LEFT    S--DOWN    D--RIGHT    0--EXIT\n");
    printf("Score:%d Move:%d\n",score,move);
    printf("Made by Yanjisheng\n");
    printf("|-------------------|\n");/* Show horizontal separator */
    for(i=0;i<=3;i++)
    {
        for(j=0;j<=3;j++)
        {
            if(code[i][j]==0)
            {
                printf("|    ");/*0 show spaces */
            }
            else
            {
                printf("|%4d",code[i][j]);/* Display numbers and dividers */
            }
        }
        printf("|\n|-------------------|\n");/* Show horizontal separator */
    }
}

int add(void)/* Intermediate variable array processing */
{
    int i;
    int t=0;
    int change=0;/* Determine whether the array has changed, 0 unchanged, a change */

    do
    {
        for(i=0;i<=3;i++)
        {
                if(temp[i]==0)
                {
                        if(temp[i]!=temp[i+1])
                                change=1;/* When a 0 instead of 0:00 behind changing array */
                        temp[i]=temp[i+1];
                        temp[i+1]=0;
                }
        }/* Remove the middle 0*/
        t++;
    }while(t<=3);/* Repeatedly */
    for(i=1;i<=3;i++)
    {
        if(temp[i]==temp[i-1])
        {

                if(temp[i]!=0)
                {
                        change=1;/* When the same two nonzero sum        array is changed, the number
of */
                        score=score+temp[i];/*加分*/
                }
                temp[i-1]=temp[i-1]*2;
                temp[i]=0;
        }
    }/* The same number of two adjacent add */
    do
    {
        for(i=0;i<=3;i++)
        {
                if(temp[i]==0)
                {
                        temp[i]=temp[i+1];
                        temp[i+1]=0;
                }
        }/* Remove the middle 0*/
        t++;
    }while(t<=3);/* Repeatedly */
    return change;
}

int main(void)
{
    int gameover=0;/* Determine whether the end of the game, an end, 0 continues */
    int i,j;
int change=1;/* Determine whether the change in the number of lattice, 0 unchanged */
char input;

srand((unsigned)time(NULL));/* Set the starting point for the random number */
while(gameover==0)
{
    if(change>=1)/* Add a new number changes only when the number of */
    {
        do
        {
            i=((unsigned)rand())%4;
            j=((unsigned)rand())%4;
        }while(code[i][j]!=0);
        if(((unsigned)rand())%4==0)
        {
            code[i][j]=4;
        }
        else
        {
            code[i][j]=2;/* Randomly selected to fill a space of 2 or 4*/
        }
        move++;/* Increase the number of */
    }
    print();/*Show */
    input=getch();/*Input direction */

    change=0;
    switch(input)
    {
        case '0':/* Quit */
            printf("Are you sure to exit?(y/n)");
            input=getchar();
            if(input=='y'||input=='Y')
                exit(0);
            break;

        case 'W':
        case 'w':/*上*/
            for(j=0;j<=3;j++)
            {
                for(i=0;i<=3;i++)
                {
                    temp[i]=code[i][j];/* The move to a variable number of columns in the
middle */
```

```
                    }
                    temp[4]=0;
                    change=change+add();
                    for(i=0;i<=3;i++)
                    {
                        code[i][j]=temp[i];/* The intermediate variable shift back a good deal
*/
                    }
                }
                break;

            case 'A':
            case 'a':/*left*/
                for(i=0;i<=3;i++)
                {
                    for(j=0;j<=3;j++)
                    {
                        temp[j]=code[i][j];/* The move to the middle line of the number of
variables */
                    }
                    temp[4]=0;
                    change=change+add();
                    for(j=0;j<=3;j++)
                    {
                        code[i][j]=temp[j];/* The intermediate variable shift back a good deal
*/
                    }
                }

                break;

            case 'S':
            case 's':/*下*/
                for(j=0;j<=3;j++)
                {
                    for(i=0;i<=3;i++)
                    {
                        temp[i]=code[3-i][j];/* The move to a variable number of columns in
the middle */
                    }
                    temp[4]=0;
                    change=change+add();
                    for(i=0;i<=3;i++)
                    {
                        code[3-i][j]=temp[i];/* The intermediate variable shift back a good deal
*/
                    }
                }
                break;

            case 'D':
            case 'd':/*right*/
                for(i=0;i<=3;i++)
                {
                    for(j=0;j<=3;j++)
                    {
                        temp[j]=code[i][3-j];/* The move to the middle line of the number of
variables */
                    }
                    temp[4]=0;
                    change=change+add();
                    for(j=0;j<=3;j++)
                    {
                        code[i][3-j]=temp[j];/* The intermediate variable shift back a good deal
*/
                    }
                }
                break;
        }
        gameover=1;
        for(i=0;i<=3;i++)
            for(j=0;j<=3;j++)
                if(code[i][j]==0)
                    gameover=0;/* All lattice are filled game is over */

    }
    printf("Game over!\n");
    getch();
    return 0;
}
```

Appendix [2]

Before the test model	1	2	3	4	5	6	7
y	64	1024	512	512	64	512	128
	8	9	10	11	12	13	14
	1024	1024	512	512	512	256	512
	15	16	17	18	19	20	21

	64	1024	128	512	256	256	128
	22	23	24	25	26	27	28
	512	1024	512	256	128	128	128
	29	30	31	32	33	34	35
	1024	128	128	512	512	64	64
	36	37	38	39	40	41	42
	128	512	128	256	256	1024	1024
	43	44	45	46	47	48	49
	512	256	64	1024	512	256	256
	50	51	52	53	54	55	56
	256	256	521	64	1024	1025	512
	57	58	59	60	61	62	63
	512	256	128	128	128	512	512
	64	65	66	67	68	69	70
	1024	512	512	1024	512	256	256
	71	72	73	74	75	76	77
	1024	512	256	128	128	1024	512
	78	79	80	81	82	83	84
	512	512	512	512	256	256	1024
	85	86	87	88	89	90	91
	128	128	256	256	512	1024	256
	92	93	94	95	96	97	98
	512	1024	128	256	1024	512	512
	99	100					
	512	512					

First experiment	1	2	3	4	5	6	7
y	2048	1024	1024	512	2048	1024	1024
	8	9	10	11	12	13	14
	512	2048	2048	512	1024	1024	1024
	15	16	17	18	19	20	21
	1024	2048	2048	1024	512	512	256
	22	23	24	25	26	27	28
	1024	2048	2048	1024	512	1024	1024
	29	30	31	32	33	34	35
	512	512	1024	2048	2048	2048	2048
	36	37	38	39	40	41	42
	1024	512	1024	1024	1024	1024	1024
	43	44	45	46	47	48	49
	2048	2048	2048	2048	512	512	1024
	50	51	52	53	54	55	56
	1024	2048	2048	2048	2048	2048	1024

	57	58	59	60	61	62	63
	1024	512	512	1024	1024	2048	2048
	64	65	66	67	68	69	70
	2048	2048	1024	512	512	256	1024
	71	72	73	74	75	76	77
	1024	2048	2048	1024	1024	512	1024
	78	79	80	81	82	83	84
	1024	1024	1024	2048	2048	2048	2048
	85	86	87	88	89	90	91
	1024	2048	1024	1024	2048	2048	1024
	92	93	94	95	96	97	98
	2048	2048	2048	2048	2048	2048	1024
	99	100					
	2048	2048					

The third test	1	2	3	4	5	6	7
y	2048	2048	2048	2048	2048	2048	2048
	8	9	10	11	12	13	14
	2048	2048	2048	2048	2048	2048	2048
	15	16	17	18	19	20	21
	2048	2048	2048	2048	1024	2048	2048
	22	23	24	25	26	27	28
	2048	2048	2048	2048	2048	2048	2048
	29	30	31	32	33	34	35
	2048	2048	2048	2048	2048	2048	2048
	36	37	38	39	40	41	42
	2048	2048	2048	2048	2048	2048	2048
	43	44	45	46	47	48	49
	1024	2048	2048	2048	2048	2048	2048
	50	51	52	53	54	55	56
	2048	2048	2048	2048	2048	2048	2048
	57	58	59	60	61	62	63
	2048	512	1024	2048	2048	2048	2048
	64	65	66	67	68	69	70
	2048	2048	2048	2048	2048	2048	2048
	71	72	73	74	75	76	77
	2048	2048	2048	2048	2048	2048	2048
	78	79	80	81	82	83	84
	2048	2048	1024	2048	2048	2048	2048
	85	86	87	88	89	90	91
	2048	2048	2048	2048	2048	2048	2048
	92	93	94	95	96	97	98

	2048	2048	2048	2048	2048	2048	2048
	99	100					
	2048	2048					

Appendix [3]

```
F = Fit[data, {1, x}, x]
72.4167 + 0.59018 x
pd = ListPlot[data, DisplayFunction -> Identity];
fd = Plot[F, {x, 20, 700}, DisplayFunction -> Identity];
Show[pd, fd, DisplayFunction -> $DisplayFunction]
```