4-Peg Hanoi Towers Algorithm Animation Demonstration System Based on Html5

Tao Wang, Yiwen Zhou and Zhihang Liu
Department of Electronics and Information Engineering, Tongji Zhejiang College, Jiaxing 314001, Zhejiang, China
Email: tjdx16060@tjzj.edu.cn

Abstract. Differing from the previous studies on demonstration Hanoi Tower algorithm based on desktop implementation, this paper adopts cross-platform HTML5 technology, drew on the references of three-peg Hanoi tower animation on the basis of fully study from the theory and realization of four-peg Hanoi tower, designed the animation implementation of four-peg Hanoi tower, simplified animation implementation and application popularized, intuitively expresses the relationship between the optimal number of moving steps and the section score in four-peg. The implementation method can also be applied to animation demonstration of other algorithm principles, and has a high reference value.

1. Introduction
In the learning process of algorithm, learners are more likely to accept dynamic algorithm demonstration compared with traditional static algorithm demonstration (such as blackboard writing, PPT, etc.) who limited by the abstractness and dynamics of algorithm description. According to the report of “The 42nd China Statistical Report on Internet Development” from China Internet Network Information Center (CNNIC), from January to June 2018, Mobile Internet access traffic consumption reached 26.6 billion GB, up 199.6% year on year. Therefore, the development of mobile terminal and the compatibility between mobile devices has become the core of the current Internet discussion. As a relatively advanced Web technology standard at the present stage, the future development direction of the Internet, HTML5 can realize the development of interface across platforms, which makes it possible to achieve animated demonstrations of algorithms. As Table 1 shows, the HTML5 animation demonstration system adopted by HTML5 technology hold the advantage in not needing to install plug-ins, well visualization effect, lower editing requirements and so on.

Hanoi Tower algorithm, as a classic recursive algorithm, is too abstract for its algorithm idea to understand for learners. So some people use animation technology for dynamically demonstrating the process of 3-peg Hanoi tower algorithm which has obtained good effect [1-4]. Although the 4-peg Hanoi tower is only one more peg than the 3-peg Hanoi tower, but the algorithm is much more complex and abstract. In order to make it easier for learners to understand the dynamic execution process of the algorithm, this paper also uses a similar animation technology to demonstrate its principle. However, the traditional 3-peg Hanoi tower is mostly implemented by C/C++ or Flash technology which has less interactive design, modifiable and extension coverage and is complex for its implementation process and is simple for its implemented UI. Based on Html5 technology, the 4-peg Hanoi tower animation demonstration system in this paper has realized 4-peg Hanoi tower animation demonstration on the basis of the original 3-peg Hanoi tower and has discussed the recurrence relation between the process parameter ‘r’ (subdivision) and the optimal movement times dynamically through carrying out algorithm dynamic process by animation parameter adjustment.
Table 1. The Comparison of Using HTML5 Implement Algorithm Demonstration and Other Using Same Kind Technology

| Language Type | Openness | Visualization Effect | Production Cycle | mobile enablement |
|---------------|----------|----------------------|------------------|-------------------|
| Flash         | Bad      | Good                 | Long             | Yes               |
| C/C++         | Good     | Bad                  | Short            | No                |
| HTML5         | Good     | Good                 | Short            | Yes               |

Touch technology compatibility Install plug-in Program language Compile
No Bad Yes ActionScript Yes
No - - C/C++ Yes
Yes Good No JavaScript No

2. Algorithm Design

2.1. Algorithm Principle
Hanoi problem is a classic algorithm problem. 3-peg Hanoi problem means that there are three pegs named A, B, C and several dishes of different sizes. The dishes are arranged on peg-A in ascending order, and the one on the top is the smallest. Now, with the assistance of empty pillar B, all dishes are moved to empty peg-C [1]. The rules are listed as follow:

- Only one dish can be moved at a time.
- Only can move the top dish.
- No larger dish may be placed on top of a smaller dish.

3-peg Hanoi tower algorithm is: first, move the top n-1 dishes of the n dishes on the peg-A to the peg-B through the assistance of peg-C. Then, move the largest remaining dish on peg-A to the aiming peg-D. By parity of reasoning, move the dishes of peg-B to peg-C with the assistance of peg-A until the movement is completed. According to the algorithm, let T(n) be the number of steps of Hanoi tower, then [5]:

\[
T(n) = \begin{cases} 
2T(n-1) & n > 1 \\
1 & n = 1 
\end{cases}
\]

It follows that:

\[
T(n) = 2^n - 1 (n \geq 0)
\]

The movement rule of the 4-peg Hanoi tower is the same as that of the 3-peg Hanoi tower. The difference between the 3-peg Hanoi tower and the 4-peg Hanoi tower is the addition of a peg (peg-D). According to Frame-Stewart algorithm, 4-peg Hanoi tower can realize on the basis of 3-peg Hanoi tower. In the movement process, to find a optimum solution, it brought in a intermediate parameter (dissection number) ‘r’ to separate the dishes on the beginning peg to another peg (‘1-r’ and ‘r-n’(0≤r≤n)) to realizing the recalling 3-peg recursion algorithm.

2.2. Algorithm Process
The core of the algorithm principle is recursion. Understanding the algorithm principle is also helpful for learners to understand the recursion thought. Based on 3-peg Hanoi tower algorithm, 4-peg Hanoi tower separated 4-peg problem through divide-and-conquer algorithm and realized the optimal solution of moving step. For the convenient of viewers, captured the key steps in the animation and pictorial described 4-peg Hanoi tower algorithm as follows:

2.2.1. As figure 1 shows, at beginning moment, ‘n’ dishes are stacked on peg-A in ascending order from top to bottom. Now the dishes on peg-A are divided into two parts, the lower part is divided into ‘r’(0≤r≤n) dishes, and the upper is divided into ‘n-r’ dishes;

2.2.2. As figure 2 shows, let the moving step of implementing ‘i’ (0≤i≤n) dishes moving for 4-peg Hanoi tower algorithm equals to F(i), and the number of moving steps required by 3-peg Hanoi tower
algorithm to realize the movement of 'i' dishes equals to T(i); Then with the assistance of peg-C and peg-D, the part of dishes ‘n-r’ on peg-A is moved to peg-B, and the number of moving steps is F(n-r).

2.2.3. As figure 3 shows, noted that since peg-B has stacked the smallest dishes, peg-B cannot be stacked any bigger dishes and there only peg A, C, D can move. Therefore, the lower part ‘r’ dishes on peg-A are moved to peg-D with the assistance of peg-C by the 3-peg Hanoi algorithm. The moving steps are: 

\[ T(r) = 2^r - 1 \]

2.2.4. As figure 4 shows, now the lower part ‘r’ dishes on peg-D have been moved, so move the upper part ‘n-r’ dishes on peg-B to peg-D according to the rules. 4-peg Hanoi tower algorithm was used to move the ‘n-r’ dishes on peg-B to peg-D with the assistance of peg-A and peg-C, and the number of moving steps equals to F(n-r). So far, the moving was completed. The total number of moving steps is:

\[ F(n) = F(n-r) + T(r) + F(n-r) = 2F(n-r) + 2^n - 1 \]

2.3. Code Design

While F(n) is the optimal solution, the subproblem F(n-r) must also be the optimal solution. So there have a method of replacing F(n-r) with F'(n-r) and produces a better solution than F(n) which is inconsistent with F(n) being the optimal solution. Therefore, this problem has the optimal substructure property and can also be solved by dynamic programming algorithm. In the moving process, ‘r’ is the undetermined value, but because of 1≤r≤n, so when r=r', namely (0≤r'≤n), F(n) is the optimal solution. In this paper, r' is the partition value of the optimal solution for F(n). In conclusion, it can be concluded that the relation between the number of moving steps of the 4-peg Hanoi tower and the parameter r is as follows:

\[
F(n) \begin{cases} 
1 & n = 1 \\
\min_{1 \leq r \leq n} [2F(n-r)+T(r)] & n \geq 1 
\end{cases}
\]

\[ T(r) = 2^r - 1 (0 \leq r \leq n) \]

As table 2, it shows the relationship between the optimal moving steps and the optimal parameter r under the condition of different number of dishes in the 4-peg Hanoi tower.
Table 2. The relation between the optimal moving steps F(n) and the optimal parameter r in the different case of dishes

| Dish n | Actual r | Actual optimal steps F(n) | Dish n | Actual r | Actual optimal steps F(n) |
|--------|----------|--------------------------|--------|----------|--------------------------|
| 1      | 1        | 1                        | 11     | 4        | 65                       |
| 2      | 1        | 3                        | 12     | 4        | 81                       |
| 3      | 2        | 5                        | 13     | 4        | 97                       |
| 4      | 2        | 9                        | 14     | 4        | 113                      |
| 5      | 2        | 13                       | 15     | 5        | 129                      |
| 6      | 3        | 17                       | 16     | 5        | 161                      |
| 7      | 3        | 25                       | 17     | 5        | 193                      |
| 8      | 3        | 33                       | 18     | 5        | 225                      |
| 9      | 3        | 41                       | 19     | 5        | 257                      |
| 10     | 4        | 49                       | 20     | 5        | 289                      |

According to the conclusion and combined with the recursive idea, it obtained the algorithm as follow:

```javascript
Hnt.start = function (n){ // Hanoi tower recursion algorithm begin
    Hnt.hanoi4 ( n, Hnt.A, Hnt.B, Hnt.C, Hnt.D );
};
Hnt.hanoi3 = function (n, a, b, c){//3-peg Hanoi tower algorithm
    if(n >= 1){
        Hnt.hanoi3(n-1, a, c, b);
        Hnt.movie(a,c);
        Hnt.hanoi3(n-1, b, a, c);
    }
}
Hnt.hanoi4 = function (n, a, b, c, d){
    let i, r, k;
    if(n == 1)Hnt.movie(a,d);
    else{
        for (i = 2, r = 0, k = n; k > 0; i++, r++)
            k = k-i;//According to the experimental results
        Hnt.hanoi4(n-r, a, c, d, b);
        Hnt.hanoi3(r, a, c, d);//3-peg Hanoi tower
        Hnt.hanoi4(n-r, b, a, c, d);
    }
}
```

3. Animation Demonstration System Design

3.1. Animation Technology Introduction

In this system, the design of data structure and algorithm, the response of interactive operation, and the drawing of graphics are all integrated and developed by HTML + CSS + JavaScript technology [6]. HTML5 is the fifth major revision of the worldwide web's core language and the applied hypertext markup language (HTML) which has good support for new CSS3 features. CSS (Cascading Style Sheets) contains basic code blocks of element reset, page layout, grid layout, form styles, general rules which manage the style and layout of page content to simplify and improve the efficiency of web front-end development. JavaScript is a client-side scripting language based on object and event driver and with relative security. It is also a scripting language widely used in client-side Web development. It only relies on the browser but does not rely on the machine and has traits of simple, efficient, safe,
high stability and strong compatibility. React is a JavaScript library for building user interfaces, primarily UI, with high productivity and flexibility. It does not need to record how the current dish moves, as single data driven, back-end data get the position of each dish to render the front animation layer through 4-peg Hanoi tower algorithm.

3.2. Animation Parameter Definition
The framework of the animation interface and the design definition of related parameters need to be considered at the beginning. A reasonable clear framework and data structure will pave the way for the design of program function modules with half the effort. In order to make the animation movement both accurate and enjoyable, the appropriate basic parameter variables such as column coordinates, dishes coordinates and dishes number were defined at the initial moment. Defined four pegs (denoted as A, B, C, D) as the array data structure, which is used to record the current storage situation of each column dishes. At the initial moment of animation, all of the dishes were put on peg-A, and the storage condition of dishes on other pillars (B, C, D) is empty. Since this animation involves the discussion on the optimal number of moving steps, and according to the discussion in the algorithm, the number of moving steps is related to the intermediate parameter r, so the intermediate parameter r and the number of moving steps in the dish must also be retained for subsequent observation and discussion.

3.3. Function Module
In the 4-peg Hanoi tower animation, the overall animation movement of the 4-peg Hanoi tower can be decomposed into multi-step single dish movement, so the single-step dish movement in the movement process is the core of animation. The one-step movement includes not only the initial position change of the dish, but also the demonstration path changes of the moving process. The coordinates of the dish are obtained by the coordinate record function. The starting position of the dish is changed by 4-peg Hanoi tower algorithm and is encapsulated in the move () function. The demo path change of the moving process is implemented by the show () function:

3.3.1. Location Record. The key to animation is position movement, each dish's data changing is a single step of its front-end coordinates. When the back-end data was updated, the front-end completes the change of interface animation through the corresponding rendering. The abscissa of the dish is the coordinate of the column, which was defined earlier. The ordinate data is recorded in the getArrCurrTop() function to implements the ordinate record. The dish triggers the movement event, that is, the movement function move () which is called to complete the single-step data change. The changed ordinate data is recorded in the getArrNextTop() function. Front end realize the mobile rendering of animation according to the function of return values.

3.3.2. Moving Algorithm. The early algorithm design has fully demonstrated the core principle of the 4-peg Hanoi tower algorithm. It implements the function in the move () function according to 4-peg Hanoi tower algorithm. When the move event is triggered, the move () function is called to update the column array where the current dish is located, and then the position record function getArrNextTop() is updated. The updated dish is stored in the new column array.

3.3.3. Path to The Mobile. As figure 2 shows, at the moving process of single dish, because of existing multiple pegs and there are multiple possibilities for changing the path of the dish. So it need to a categorical discussion of the various situations:
  - If there is no dish on the initial peg (A), return.
  - If it is the movement between adjacent pegs (that is, A and B, B and C, C and D), the movement can be realized according to the initial position and the end position of the dish. The movement process includes horizontal movement, upward movement and downward movement.
• If it is the movement between two pegs of peg-A (that is, A and C, B and D), the position of the highest dish of the middle peg (B or C) can be obtained and the path can be moved from this position.
• If it is the movement between two pegs (that is, A and D), the highest dish position of the middle peg (B and C) can be obtained and the path can be moved from this position. The moving process is divided into horizontal movement, upward movement and downward movement.

Figure 2. An example of the middle mobile movement of a 4-peg Hanoi tower

4. System Implementation Effect

Figure 3. 4-peg Hanoi tower algorithm graphical presentation interface

As shown in figure 3, the interface is divided into animation demonstration area, parameter setting area, parameter statistics area and game control area.

Users can adjust animation parameters by themselves and control the play during the animation process. Interactive animation was also used to modify the intermediate parameter ‘r’ that affects the number of moving steps, so readers can easily understand the meaning of its parameters.

4.1. Animation Demonstration
Carrying on the 4-peg Hanoi tower animation broadcast.

4.2. Parameter Setting
Including animation parameter Settings: dishes number, animation sound effects, auto-play, drag speed bar.
• After users input legitimate dishes number, select the parameter option and click the start button in the game control area at the bottom right corner to start the animation.
• When users select parameter settings selectively, click the right button to start and the animation will play automatically until the end.
• The user can move the speed bar when the animation is played or before to achieve the best matching speed.
4.3. Game Control
Provided start, pause, forward, backward, replay, and exit buttons. If the user wishes to play the animation frame by frame, cancel the check of the auto-play button and select the "start" button before starting the animation. Click the "forward" and "backward" buttons in the control area to control the animation automatically. When the animation is finished, the user can click the "replay" button to watch the animation again and click the exit button to close the page.

4.4. Parametric Statistics
Displayed the relevant parameters of the animation, including plates number, moving steps, and score.

5. Conclusion
The study of Hanoi tower problem is not only limited to make learners understand its core algorithm, but also plays an important guiding role in practical application. For example, the relation between the optimal movement times and the movement scheme is of guiding significance to the design of memory optimization algorithm. The recursive relationship between the optimal number of moves and the movement scheme is analyzed, and the results obtained from the recursive relationship can be further analyzed in such analysis types as optimization analysis and satisfaction analysis. Recursive relation model has a very wide application background, such as ecological model, cobweb model in market economy, infectious disease model, population control model.

In this paper, the 4-peg Hanoi tower as an example and has achieved the algorithm animation visual demonstration system which has the characteristic as follow:
- Programming implementation is relatively simple.
- Free interface design degree, high secondary scalability.
- Interactive. Have interactive editing and increased the fun of learning from the original can only watch animation to understand the principle of the algorithm
- Access the page system through the browser, allowing visitors to switch between different mobile devices at any time, such as notebook, mobile phone, tablet, etc.
- Lower cost of access, convenient for later maintenance and update, higher promotion.

6. References
[1] HUANG Jun. On the Problems of 4-peg Hanoi Tower of C Language [J]. Journal of Qiannan Normal College of Nationalities, 2012(06):79-82.
[2] WEI Hong-chun. Demo in Hanoi Tower in Graphical environment[J]. Electronic Design Engineering, 2014(15): 8-10+14.
[3] DU Hai-long. Hanoi Tower Animation Demonstration System Based on Recursive Algorithm [J]. Electronic Technology and Software Engineering,2016(24):98-99.
[4] LI Jian. Design and Implementation of Hanoi Tower Algorithm Animation System [J]. Modern computer (professional),2011(24):76-80.
[5] YANG Kai, XU Chuan. The Preliminary Probe of 4-peg Hanoi Tower [J]. Acta Scientiarum Naturalium Universitatis Pekinensis,2004(01):99-106.
[6] ZHAO Jian, Research and Implementation of HTML5 Animation Engine Technology [D], Beijing: Beijing University of Posts and Telecommunications. 2015:7-20.
[7] DAI Song, XU Ran, ZHOU Zhong. HTML5-Based Algorithm Animation Online Platform [J]. Journal of System Simulation,2013(10):2436-2448.
[8] FU Jin-zhi, HUANG Shi-mei. Design and Implementation of Data Structure Algorithm Demonstration System Based on HTML5 [J]. Laboratory science,2015(02):72-75.
[9] LI Xiao-hong, LIU Cong. LUO Jia-wei. Review of Algorithm Visualization Systems: A Learner Perspective [J]. Computer Science, 2015(S2):431-437+453.
[10] PING Shu-wen, PAN Jue-yu, ZHANG Xue-jin, DU Xiao-rong. Development of a Lightweight Animation Engine Based on HTML5 and JavaScript [J]. Computer Technology and Development,2013(12):5-10.