MedCrave
Step into the World of Research

International Robotics & Automation Journal

Research Article

Open Access

CrossMark

Optimal solutions for lightbot junior version and new map designs

Abstract

Lightbot is an interesting game app that has strong educational purpose to guide kids at young age to learn programming skills. It has a free version named Lightbot Code Hour consisting of 20 levels and a paid version with 50 levels that we studied in the past. It also has a junior version with 42 levels for young kids. Although it is not hard to solve the puzzles, finding the optimal solutions with the smallest number of blocks is non-trivial for some levels. In this paper, we provide a complete list of the block numbers for optimal solutions in the junior version, as well as the detailed analysis of all the non-trivial levels. The overall optimal number for 5 stages is found to be 361. We also make new map designs to create complicated solutions with many periods.

Keywords: Lightbot, optimal solution, periodicity

Volume 12 Issue 1 - 2026

Songming Hou,[1] Jianning Su,[2] Yin Sun[3]
[1]Program of Mathematics and Statistics and Center of Applied Physics, Louisiana Tech University Ruston, USA
[2]Department of Mathematics, Computer Science, and Engineering, Georgia State University Perimeter College Georgia
[3]Mathcup Academy, USA

Correspondence: Songming Hou, Program of Mathematics and Statistics and Center of Applied Physics, Louisiana Tech University Ruston, Louisiana, 71272, USA

Received: September 18, 2026 | **Published:** October 01, 2026

Introduction

Over 30 or 40 years ago, most kids did not have computer access daily.[1,2] The situation has completely changed. With children now having widespread access to computers and other electronic devices, educational games like Lightbot.[3] have become valuable tools for introducing fundamental programming concepts such as loops and subroutines. Serious-game designs have drawn quite some attentions.[4–6] Lightbot has a free version named Lightbot Code Hour consisting of 20 levels and a paid version with 50 levels, also a paid junior version with 42 levels. The goal is to light up all the lights using a robot by giving it fixed instructions. It has been studied in.[3,7] However, optimal solutions are not discussed until the recent work.[8,9] Optimal solutions found add up to 173 blocks after a rigorous check by the computer for the free version. The paid version was not systematically studied in the literature until our recent work.[10,11] Optimal solutions found add up to 471 blocks.

When looking for optimal solutions for Lightbot levels, the idea of periodicity comes into play which is shared by another game named the Rubik's Snake.[12] Some mathematical problems including periodicity concerning a Rubik's Snake have been studied.[13,14] A comic book was also published to make some basic research accessible for kids.[15]

We borrow some ideas from the study of Rubik's snake to carefully study the optimal solutions for Lightbot junior version and provide a complete list of the block numbers for optimal solutions, as well as the detailed analysis of all the non-trivial levels.

The rest of the paper is organized as follows: in Section 2, we give a complete list of number of blocks needed for the junior version. In Section 3, we explain the non-trivial levels in details. In Section 4, we give new map designs for complicated solutions with many periods. We conclude in Section 5.

A complete list of the number of blocks needed for optimal solutions

In this section, we provide a complete list of the number of blocks for optimal solutions. All these optimal solutions are verified by our computer code that they do solve the problems and they cannot be further improved. Players can use the information to compare with their own solutions.

For Stage 1 levels, the number of blocks are 3+5+9+6+12+11+6+11+9=72.

For Stage 2 levels, the number of blocks are 4+4+6+6+10+11+12+12+10=75.

For Stage 3 levels, the number of blocks are 5+6+10+11+4+10+10+11+12=79.

For Stage 4 levels, the number of blocks are 8+7+11+9+13+11+14+17=90.

For Stage 5 levels, the number of blocks are 4+4+8+8+8+7+6=45.

Overall, the verified optimal solution is 72+75+79+90+45=361. See Figure 1.

Figure 1 The overall optimal solution for Lightbot junior version.

Non-trivial levels for junior version

There is no need to explain the first three stages, as they are all sequential with no subroutine allowed. We will explain the non-trivial levels 4-3, 4-7, 4-8 and 5-7.

The idea for getting an optimal solution for 4-3 is, clearly "jump, light, jump, light, jump, light" is a pattern that can be applied to

©2026 Hou et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and build upon your work non-commercially.

Copyright:
©2026 Hou et al.

both stairs. We just need to modify this in the subroutine to ensure it also works for the connection part between stairs, taking advantage of the fact that we can make a light off then on again. Figure 2 shows an optimal solution. It is easy for human players to overlook the optimal solution for level 4-7 because the common idea is to sweep column by column. The trick here is to establish a spiral sweep for two rows then jump up two levels and repeat. Figure 3 shows an optimal solution.

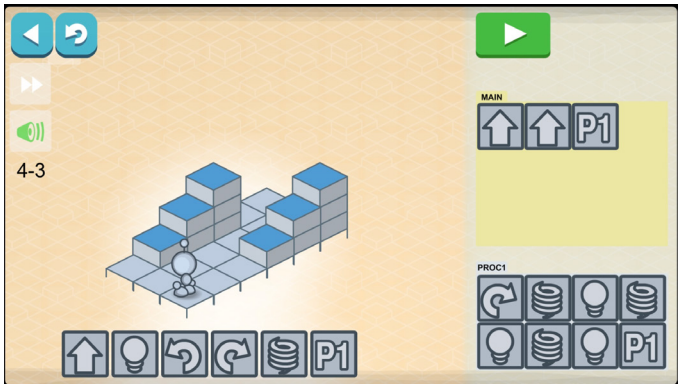

Figure 2 Level 4-3.

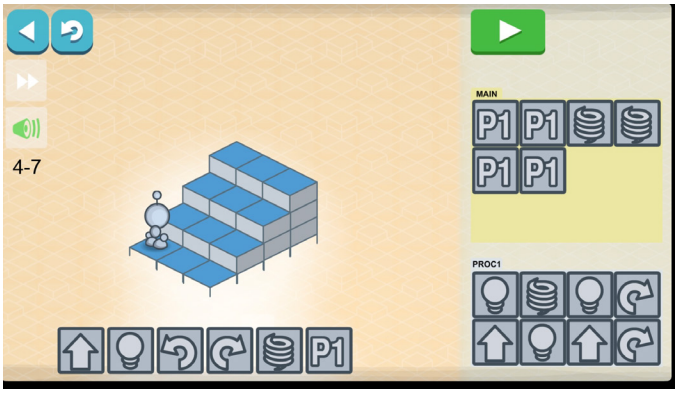

Figure 3 Level 4-7.

In level 4-8, the idea is similar to 4-3 that we use a common sequence to handle the straight line parts and connection parts. Figure 4 shows an optimal solution.

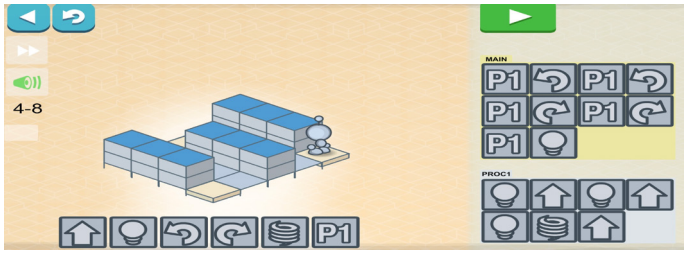

Figure 4 Level 4-8.

In level 5-7, the idea is to come up with a period that, when repeating several times, the robot goes to a symmetric position with a 90 degree rotation then we could repeat. Figure 5 shows an optimal solution.

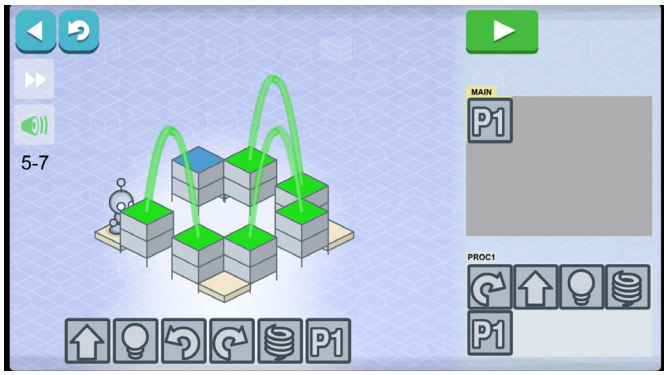

Figure 5 Level 5-7.

The optimality of all levels are rigorously verified as in.[9–11] The idea is to exhaust all combinations with small number of total blocks, and for each combination, terminate the search if all lights are on or if the current state (a combination of robot position, robot orientation, elevator height and light status) has been seen in the history.

New designs for complicated solutions with many periods

We would like to extend beyond the existing levels. The third author has created a website

https://ed2k.github.io/Lightbot/src/develop/ by modifying existing open source codes. The new functions added include a map editor (the orange button) that allows us to modify the map for our research purpose. Also, two keys "fast" and "fastest" are added in addition to "run" so that the execution speed can be adjusted to handle a solution with many periods. The implementation of elevators is also added by the third author. Figure 6 shows a new design of the map using the website. Figure 7 shows the map editor. (The initial location and orientation of the robot can be changed in the main program by making initial moves.) The interesting part is that readers can all access the website for free and design their own maps then solve their own puzzles.

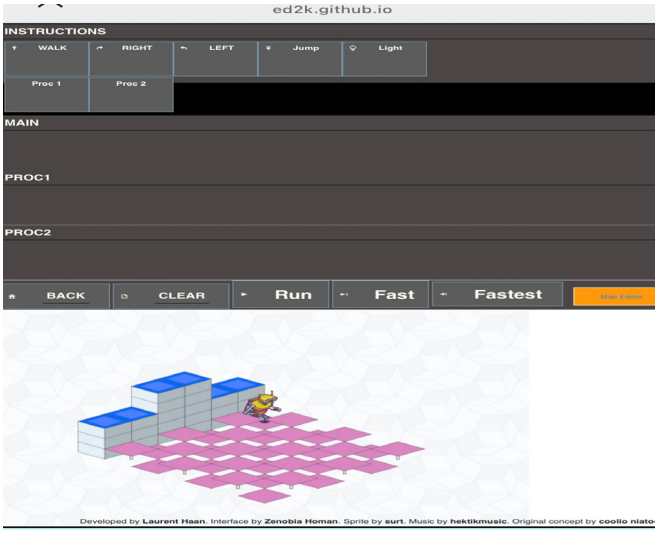

Figure 6 New map design.

Citation: Hou S, Su J, Sun Y. Optimal solutions for lightbot junior version and new map designss. *Int Rob Auto J.* 2026;12(1):122–124.
DOI: 10.15406/iratj.2026.12.00324

Copyright: ©2026 Hou et al.

Figure 7 New map editor.

The main idea to generate solutions with many periods is as follows. First, we need several lights so that it is not easy for them to be on simultaneously. Second, we need many elevators cause the periodic 3 variation of an elevator combined with periodic 4 rotation can generate many variations. Third, we place elevators in an alternating pattern with even and odd heights to create further variations. Fourth, we need to pick a sequence that have enough variations. Finally, some minor adjustments can be done using exhaustive search. Note that the solution is not needed to have the least number of blocks for this research purpose.

We use a 6-by-7 map similar to the one in level 24 from the website. (1,1) is the lower right corner. For the first 6-by-6, the elevators are at height 0 for even sum of indexes and height 1 for odd sum of indexes to start with. The 7th column have 6 lights at some heights. The robot is set to be at (6,6) location with the column of lights on its back. The sequence we propose is "forward, light, jump, jump, light, jump, left". Tap to highlight "MAIN". Tap "Proc1" in instruction list to let MAIN program call PROC1. Tap to highlight the "PROC1" below (the one with all capital letters). Then tap "forward, light, jump, jump, light, jump, left" one by one to bring them into PROC1. Finally tap "Proc1" in the instruction list another time so that it goes to the end of the list in PROC1 to call itself, forming a loop. This gives quite some variations. By an exhausting search of all possible heights up to 6, we found that the 7th columns with "6,3,3,5,5,3" heights have the most number of periods: 656547 periods! Note that the original Lightbot game, no matter paid or free version, at least for optimal solutions, the periods are all less than 50.

Now we fix the "6,3,3,5,5,3" heights for lights and make a minor adjustment to elevators. We only modify the initial heights of two elevators by 2 or 4. After an exhaustive search, if we modify the (1,5) location to have initial height of 2 and the (2,2) location to have initial height of 4, it corresponds to the most periods we found: 1767135 periods! This would take minutes to execute even using the "fastest" button to speed up but in the end, all lights are indeed on simultaneously.

To summarize, the procedure above combines heuristic idea like using elevators with alternating heights, rigorous local search like exhausting heights of the 7th column up to 6, and an overall plan to fix some conditions and exhaustively search for others each time to keep the computational cost within reach. When global exhaustive search is out of reach (the heights of blocks and elevators, the robot position and orientation, the light locations, and the sequence can all vary, much more complicated than finding optimal solution for those original lightbot versions) we can still use more than one local search within reach. Readers are welcome to verify the result by actually editing the map using the map editor and compare the time spent for each solution.

Conclusion

Although Lightbot (junior version) is a game for kids to learn programming skills, finding optimal solutions is non-trivial especially for some tricky levels of the junior version. We gave detailed optimal solutions to the non-trivial levels of Lightbot (junior version) and a complete list of conclusions verified by our computer code for all levels. Overall the optimal solution is 361. We also designed new maps to have complicated solutions with as many as 1767135 periods. Readers are welcome to design their own maps as well using the website.

Acknowledgment

We thank Daniel Zhang, Lily Jia and Mingjia Yang for some inspiring discussions and testing the game. S. Hou's research is partially supported by the Walter Koss Endowed Professorship. The title of the professorship is made available through the State of Louisiana Board of Regents Support Funds.

Conflicts of interest

None.

References

1. Households with a computer 40 years ago. 2021.
2. Households with a computer now.
3. Gouws LA, Bradshaw K, Wentworth P. Computational thinking in educational activities: an evaluation of the educational game light-bot". In Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '13, 2013. p. 10–15.
4. Akkaya A, Akpinar, Y. Experiential serious-game design for development of knowledge of object-oriented programming and computational thinking skills. Computer Science Education. 2022;32(4):476–501.
5. Arif YM, Ayunda N, Diah NM, et al. A systematic review of serious games for health education: Technology, challenges, and future directions. Transformative Approaches to Patient Literacy and Healthcare Innovation. 2024. p. 20–45.
6. de Carvalho CV, Coelho A. Game-based learning, gamification in education and serious games. *Computers*. 2022;11(3).
7. Gil A, Figueira T, Netto JFM. Extracting learning analytics from lightbot gameplay sessions. XXIII Simpo´sio Brasileiro de Jogos e Entretenimento Digital (SBGames 2024) - Manaus/AM. 2024.
8. Hou S, Huang Y. Optimal solutions for lightbot code hour. *International Robotics and Automation Journal*. 2025;11(1):8–11.
9. Hou S, Su J, Huang Y. Lightbot code hour: optimal solutions, theories and designs. *International Robotics and Automation Journal*. 2025;11(1):19–23.
10. Hou S, Su J. Optimal solutions for the first three stages of paid version of lightbot. *International Robotics and Automation Journal*. 2025;11(3):98–102.
11. Hou S, Su J. Optimal solutions for the last three stages of paid version of lightbot. *International Robotics and Automation Journal*. 2026;2(1):48–52.
12. Fenyvesi C. Rubik's snake of 'infinite possibilities. The Washington Post; 1981.
13. Hou S, Chen Y, Li Z. Some mathematical problems related to the rubik's snake. *J Mech Rob*. 2021;13(1):014502.
14. Hou S, Su J, Chen Y. Palindromic, periodic and möbius rubik's snakes. *International Robotics and Automation Journal*. 2021;**7**(3):84–88.
15. Pan E, Hou S. *The Legend of the Rubik's Snake*. Lit House LLC. 2025.

Citation: Hou S, Su J, Sun Y. Optimal solutions for lightbot junior version and new map designss. *Int Rob Auto J*. 2026;12(1):122–124.
DOI: 10.15406/iratj.2026.12.00324