General Solutions to the Lost Riches Light-Up Tile Puzzles

arisu
twitter.com/aris_uu

geq0
twitter.com/GreaterOrEqual0

Abstract

This paper describes the light-up tile puzzle from the game Genshin Impact. Given a 3×3 grid of tiles and a light configuration, the objective of the puzzle is to find a walk to light up the tiles based on the configuration. The tiles will toggle its light state when stepped on. In this paper, we proved that any configuration in an arbitrarily sized grid has a solution. In particular, we constructed a general algorithm to solve any configuration of any size.

1. Introduction

1.1. Light-Up Tile Puzzle

This puzzle in Genshin Impact consists of tiles that light up or turn off when a player steps on them. There are different variations to the puzzle: some require the player to light up all the tiles without stepping on any tile twice, and some require the tiles to be lit up in a specific configuration [1]. In this paper, we focus on a specific variation where the player must light up certain tiles according to a given configuration in a 3×3 grid. All tiles are initially turned off, and the player starts on the bottom right tile (immediately toggling it). The player can move up, down, left or right to an adjacent tile, and the light state of the destination tile will be toggled (between on and off). This variation of the puzzle existed as a limited time event, Lost Riches 2021-08-06 [2].

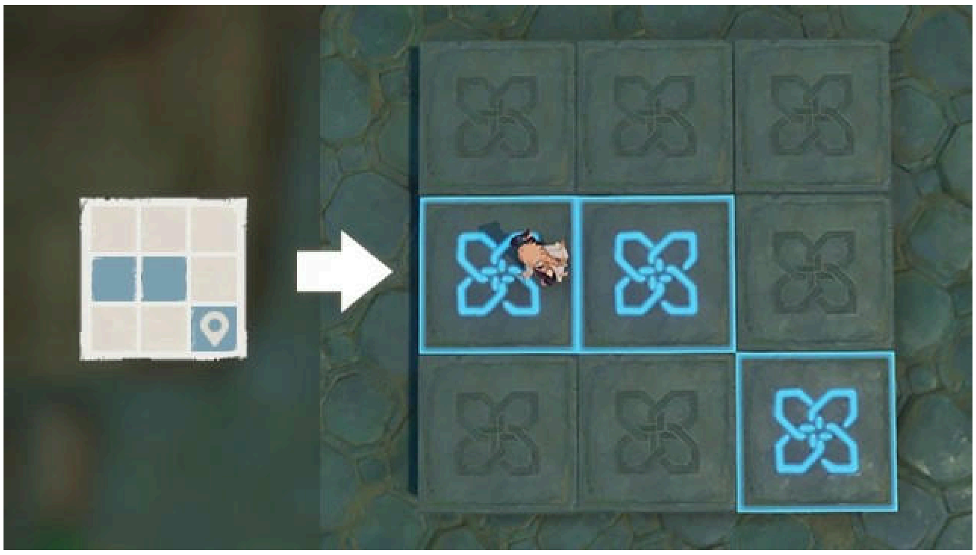

Figure 1: Example puzzle with a 3×3 grid

In this paper, we will prove that every 3×3 puzzle configuration has a solution. Moreover, we will construct an algorithm that allows us to solve the light-up tile puzzle for any arbitrary configuration in an $m \times n$ grid.

1.2. Graph Theory Interpretation

It turns out that this problem can be represented in terms of graph theory. First, we define the corresponding grid as a graph.

Definition 1.2.1: A *grid graph* is a graph $G(m, n)$ consisting of $m \times n$ vertices arranged in a grid with edges connecting vertices with the difference of its indices equal to 1.

$G(m, n) = (V, E)$ where

$$V = \{v_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$$
$$E = \{\{v_{i,j}, v_{i,j+1}\} \mid 1 \leq i \leq m, 1 \leq j < n\} \cup \{\{v_{i,j}, v_{i+1,j}\} \mid 1 \leq i < m, 1 \leq j < n\}$$

Notation: The vertex of a grid graph at row i and column j is written as $v_{i,j}$.

Definition 1.2.2: A *puzzle configuration* is a matrix V_G with entries 0 or 1 where 1 means the tile is lit up and 0 otherwise.

The entries in V_G corresponds to the vertices in the natural way.

Because of this, a solution to a configuration can be represented as a walk. For the sake clarity, the definition of walk is provided below.

Definition 1.2.3: A *walk* is a sequence of vertices (can be repeated) of a graph where each adjacent vertices in the sequence are connected by an edge.

Definition 1.2.4: A *multiset* is a set in which duplicates are allowed. The number of occurences an element has in a multiset is called its *multiplicity*.

Now, the original problem can be rephrased as the following equivalent problem.

Problem 1.2.1: Given a *puzzle configuration*, find a walk that starts at the starting point (lower-right corner) such that the multiplicity of the vertices in the multiset of the walk's vertices are congruent to the puzzle configuration in modulo 2 (has the same parity).

Example:

The puzzle configuration shown in Figure 1 can be represented as the matrix

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

A possible solution to the above puzzle is the sequence

$$\left(v_{3,3}, v_{3,2}, v_{2,2}, v_{2,1}, v_{3,1}, v_{3,2}, v_{3,1}\right)$$

Now we are ready to state our main result.

Theorem 1.2.1 (Main Result): Any puzzle configuration of any size has a solution, that is for every m, n, and a puzzle configuration V_G, there exists a walk that starts at the starting point (lower-right corner) such that the multiplicity of the vertices in the multiset of the walk's vertices are congruent to the puzzle configuration in modulo 2 (has the same parity).

2. Proof of the Main Result

First, we will prove a key lemma that will help us construct the algorithm.

Lemma 2.1 (Flip and move): Given two adjacent vertices v_1 and v_2, with v_1 as a starting point, you can move to v_2 by only flipping v_1's state using the walk (v_2, v_1, v_2).

Proof. The state of v_1 will flip because it's visited once, and the state of v_2 is not changed because it's visited twice. Moreover, your current position will move to v_2 after that walk. □

Based on this lemma, now we provide the following Algorithm 1 that works for matrices of size $1 \times n$. The idea of the algorithm is that we check if the current tile we're stepping on is already correct, then we just move to the next tile, otherwise do the "flip and move" from Lemma 2.1 and repeat until we reach the end. Formally, we may state the algorithm recursively as follows.

Algorithm 1.

```
input: a 1-indexed boolean array of length n representing the puzzle configuration
output: an array of vertices that is a solution to the puzzle
function solution(puzzle)
  k ← puzzle.length
  if k = 1 then
    if puzzle[1] = true then
      return [1]
    else
      return []
    end
  if puzzle[k] = true then
    return [k] + solution(puzzle[:k])          ▹ [:k] means slicing the array up until before index k
  else
    return [k, k-1, k] + solution(puzzle[:k-1] + [not puzzle[k-1]])
  end
end
```

Next, we will prove the correctness of the algorithm.

Theorem 2.1: Algorithm 1 is correct.

Proof (Correctness of Algorithm 1). We will prove this using induction on the size of the matrix. Let $P(k)$ be the statement "Algorithm 1 is correct for puzzles of size $1 \times k$".

$P(1)$ is trivially true by enumeration. The puzzle (1) has a solution of $(v_{1,1})$, and (0) has a solution of () (the empty walk).

For the induction step, suppose that $P(k)$ is correct for some natural number k. Now, take a puzzle configuration of length $k+1$ as an input. We then split by cases on the $(k+1)$-th tile

Then, if the $(k+1)$-th tile should be lit, the algorithm will produce the walk $(v_{1,k+1})$ followed by the solution of the puzzle with the tiles $1, ..., k$. This produces the correct solution because the multiplicity of $v_{1,k+1}$ is 1, and the other tiles are guaranteed to be correct by $P(k)$.

Else, if the $(k+1)$-th tile should not be lit, the algorithm will produce the walk $(v_{1,k+1}, v_{1,k}, v_{1,k+1})$ followed by the solution of the puzzle with the tiles $1, ..., k$ with the k-th tile flipped. Then, $v_{1,k+1}$ will have multiplicity 2 which is even (off), $v_{1,k}$ will have its state flipped twice, so it returns to what it should be, and the remaining tiles will be correct by $P(k)$.

Therefore, Algorithm 1 is correct for all puzzles of size n. $\square$

Now, we are ready to prove the main result. We may recall the statement of the result.

Theorem 1.2.1 (Main Result): Any puzzle configuration of any size has a solution, that is for every m, n, and a puzzle configuration V_G, there exists a walk that starts at the starting point (lower-right corner) such that the multiplicity of the vertices in the multiset of the walk's vertices are congruent to the puzzle configuration in modulo 2 (has the same parity).

Proof. For a $m \times n$ grid, we can transform it into a $1 \times mn$ grid in a "snake-like" manner starting from the lower-right corner. If m is odd, the resulting grid will be $(v_{1,1}, ..., v_{1,n}, v_{2,n}, ..., v_{m,1}, ..., v_{m,n})$. Else, if m is even, the resulting grid will be $(v_{1,n}, ..., v_{1,1}, v_{2,1}, ..., v_{m,1}, ..., v_{m,n})$. This transformation has the property that adjacent nodes in the resulting grid is also adjacent in the original grid. But then Algorithm 1 provides a solution for grids of this size. So we are done. $\square$

3. Final Remark

We note that with some further arguments, this result can be extended to any weighted connected graphs with Hamiltonian path. However, for the sake of argument clarity, we only provided our proofs for the case of grid graphs. On the other hand, there are still many possible developments of this problem. For example, it is certainly an interesting question to count the shortest walk possible on any arbitrary configurations. We invite interested readers to further explore these kind of problems.

Acknowledgement

We thank Muhammad Afifurrahman (UNSW) for suggestions for improving this paper's draft.

References

[1] "Light-Up Tile Puzzle". [Online]. Available: https://genshin-impact.fandom.com/wiki/Light-Up_Tile_Puzzle

[2] "Lost Riches 2021-08-06". [Online]. Available: https://genshin-impact.fandom.com/wiki/Lost_Riches/2021-08-06