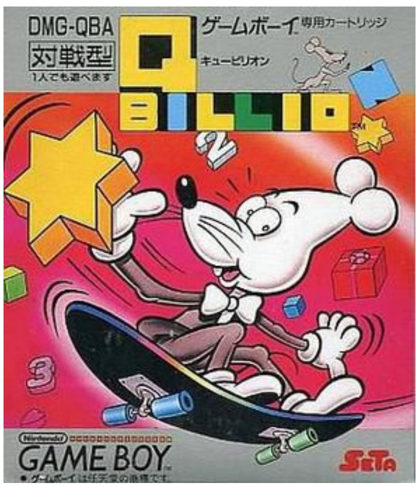
Q-Billion is NP-Complete

Zack Romrell and Yuan Qiu

November 2022

1 Introduction

In the following section, 1.1, we discuss the objective and rules of the classic Gameboy game Q-Billion. In 2.1 we add a couple simplifying restrictions and an extra feature to Q-Billion. In 2.2 we establish a reduction to a modified version of HPGRID (Hamiltonian Path in Grid Graph), HPGRID', then establish a reduction from HPGRID' to Q-Billion. In 2.3 we discuss some of the difficulties that occurred while trying to prove membership in NP. Lastly in 3 we discuss a possible approach to proving membership in NP [1].



1.1 Objective and Rules of Game

Q-Billion is a game that requires the player to control a small animated mouse character that can move in the standard cardinal directions: up, down, left and right. On the Nintendo game boy, Q-Billion is typically played on a 9×10 grid, however, for the classification of a complexity class we consider the game on an arbitrarily large grid. The game is displayed from the bird's eye perspective and the objective of the game is to push all boxes so that they are no longer stacked (i.e have a height of 1). Each box/stack of boxes takes up an individual cell and has a label on top corresponding to the total height of the stack. A box with a label of 3 corresponds to 3 boxes stacked on top of each other. The mouse character is allowed to travel to each grid cell no matter how many boxes are stacked, however, you are only able to push an adjacent block that is at the top of a stack and is exactly 1 unit higher than the current height of the block you reside on. If there is another block in the path you are trying to push, you unfortunately are not strong enough to push 2 blocks at once. Many difficulties arise in the game due to the fact that you can only push and not pull a block. This allows for one to push a block into a dead zone where it is unable to be moved for the rest of the round. Also, the limitation of only being allowed to push a block at the top of its stack of height one greater than your current height requires the level to have a built in staircase for any stack of height greater than 2. The blocks also can fall an arbitrary distance down when pushing them and the corresponding labels of the heights update accordingly. As one reaches higher levels, blocks that have a design are introduced. For the purpose of our paper we do not consider the complexity when this larger class of blocks are introduced [2].

2 NP-Completeness

Recall that in order to establish a problem is NP-complete one must show that the problem is both at least NP-hard and a member of NP. A decision problem, D, is at least NP-hard if every problem in NP can be polynomial time reduced into it. Therefore, it is enough to provide a polynomial time reduction from a known NP-hard problem to D. In terms of showing containment in NP one can approach this using a decision tree, a guessing algorithm, or a certificate verification algorithm.

2.1 Simplifying Assumptions

As described in 1.1 there are a lot of different rules and small nuances that can differentiate between successfully completing a level versus not. As previously mentioned we do not consider blocks that have a specific pattern on them. Also, in our analysis we found that the ability for the mouse to jump to a cell of arbitrary height presents a lot of issues when trying to derive a reduction from other NP-Complete problems. Therefore, in our analysis we restrict the mouse to only be able to travel to a cell of height at most 1 greater than the current cell's height. Note that the mouse character is still allowed to jump down from a cell of arbitrary height, however, this doesn't serve as a useful application in our NP-hard proof. In our NP-hard reduction we also introduce immovable blocks that are separate from the game's objective (i.e are not required to be pushed down to a height of 1). We utilize these blocks to allow for a more creative level structure. With these two assumptions we are ready to dive into our reduction and analysis.

2.2 NP-Hardness

In this section we will provide a polynomial time reduction utilizing the source problem HPGRID which was proven to be NP-Complete, hence NP-hard, in [3]. HPGRID accepts an input of a grid graph, $G[P] = (P, E)$ where $P \subseteq \mathbb{Z} \times \mathbb{Z}$ and vertices $s, t \in P$ and outputs True if $G[P]$ has a Hamilton path from s to t and False otherwise. However, as you will see later our reduction to the Q-Billion version described in 2.1 presents difficulties when trying to specify the end vertex of the Hamilton path. Thus, we will first establish that HPGRID' is NP-hard which we will define below. Note going forward we will refer to the modified version of Q-Billion described above as simply Q-Billion.

Definition 2.1. HPGRID' is a Hamilton path problem that accepts an input of a grid graph, $G[P] = (P, E)$ where $P \subseteq \mathbb{Z} \times \mathbb{Z}$ and vertex $s \in P$ and outputs True if $G[P]$ has a Hamilton path from the start vertex s and False otherwise.

Theorem 2.2. *HPGRID' is atleast NP-Hard*

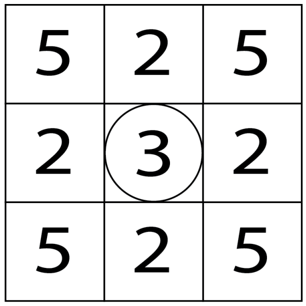
Proof. Consider the following polynomial time reduction from HPGRID to HPGRID'. Take the input graph, $G[P] = (P, E)$, for HPGRID with specified $s, t \in P$ and build $|t|$ grid subgraphs where $|t|$ represents the valency of vertex t and in each grid subgraph leave exactly one of t's edges. There will be $G_1[P], \ldots, G_{|t|}[P]$ resulting subgraphs each of which have a leaf namely t. Run HPGRID' on each of the produced subgraphs (there can be a minimum of 2 grid subgraphs and a maximum of 4) and return the logical or of all the outputs (i.e if one returns True, the overall output is True). Note that the following reduction is polynomial time since it requires making at most 4 subgraphs of $G[P]$. In order to verify the following reduction is correct we must show that HPGRID($G[P], s, t$) = True if and only if HPGRID'($G_1[P], s$) $\vee \ldots \vee$ HPGRID'($G_{|t|}[P], s$) = True.
Direction 1: HPGRID($G[P], s, t$) = True $\implies$ HPGRID'($G_1[P], s$) $\vee \ldots \vee$ HPGRID'($G_{|t|}[P], s$) = True. If HPGRID($G[P], s, t$) = True, then let M denote the Hamilton path in $G[P]$ from s to t. Let M_i' be the same path in $G_i[P] \, \forall i \in \{1, \ldots, |t|\}$ except for the last edge to vertex t and vertex t. Note that since M traverses one edge $(?, t)$ to get to the last vertex in the path, t, the same path from ? to t will only be available in one of the grid subgraphs $G_i[P]$. The subgraphs that do not have the correct edge will output False, while the subgraph, $G_k[P]$, that does have the corresponding edge will output True. Note that M is a Hamilton path on $G_k[P]$ since $G[P]$ and $G_k[P]$ share all the same vertices and also share the last corresponding edge from the second to last vertex, ?, to t. Therefore, HPGRID'($G_1[P], s$) $\vee \ldots \vee$ HPGRID'($G_{|t|}[P], s$) = True.
Direction 2: HPGRID'($G_1[P], s$) $\vee \ldots \vee$ HPGRID'($G_{|t|}[P], s$) = True $\implies$ HPGRID($G[P], s, t$) = True. If HPGRID'($G_1[P], s$) $\vee \ldots \vee$ HPGRID'($G_{|t|}[P], s$) = True, then HPGRID' with at least one grid subgraph $G_1[P], \ldots, G_{|t|}[P]$ outputted True. Let M' be a Hamilton path for a single $G_k[P]$ that outputted True. Since the specified start vertex was s and t is a leaf in the grid subgraph $G_k[P]$, it must be that M' ends at t. By design each $G_i[P] \forall i \in \{1, \ldots, |t|\}$ is a subgraph of $G[P]$ only missing edges, thus it directly follows that any Hamilton path in the subgraph is also a Hamilton path in $G[P]$ since $G[P]$ contains the same vertices as $G_k[P]$ and a super set of the edges of $G_k[P]$. Therefore, HPGRID($G[P], s, t$) = True.
Thus, HPGRID' is NP-Hard. ☐

Now that we have verified that HPGRID' is at least NP-Hard a reduction from HPGRID' to Q-Billion is sufficient to demonstrate Q-Billion is also at least NP-Hard.

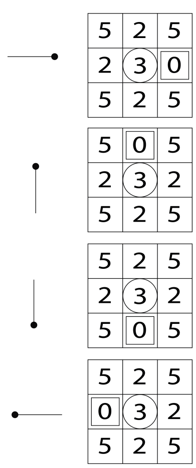
Theorem 2.3. *Q-Billion is at least NP-Hard*

Proof. Consider the following polynomial time reduction from HPGRID' to Q-Billion. First it is important to note that the mouse that the user controls in Q-Billion resides in the actual area of the grid cell of the $n \times m$ grid instead of living on the vertices and edges like a Hamilton path would. Thus, we must scale the grid graph supplied from HPGRID' in the reduction to correspond to the proper functionality intended for the Q-Billion level. Secondly, for simplicity the start vertex s that gets passed into HPGRID' is going to correspond to the upper left most vertex in the input grid subgraph, $G[P]$.

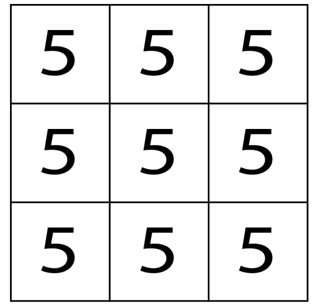
Before getting into the formal mathematical definitions consider the following visuals. Each vertex that has valency greater than or equal to 2 or is the start vertex, s, in $G[P]$ will be mapped to a following 3×3 grid of cells where each cell represents a cell in the Q-Billion level.



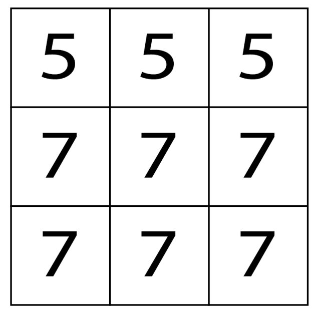
Each block with the number represents a stack of immovable blocks of height corresponding to the specified number. The block in the center with the embedded circle represents a movable block which must be pushed to the ground level (i.e a square tile of height 0) in order for the level to be completed. In the following mapping there is only one immovable block stacked on top of two immovable blocks. From the birds eye view it is ambiguous what the cell truly represents. Technically it could represent 3 movable blocks stacked, 1 immovable block followed by 2 movable, or the current case two immovable blocks followed by a movable block. This ambiguity plays a role in levels of Q-Billion that incorporate different designs on blocks. In this case one would only know the design of the topmost block in the stack. Each vertex of valency 1 in $G[P]$ will mapped to one of the following 3×3 grid depending on which cardinal direction the edge comes from. The ground level box being on the other side of the movable block in the same cardinal direction.



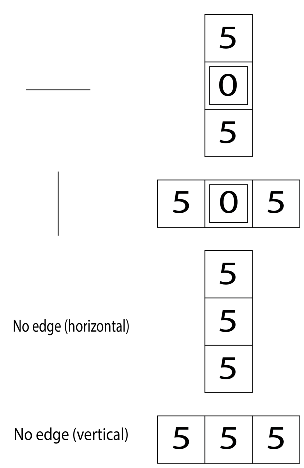
Holes in the sub grid graph (i.e missing vertexes) will be mapped to a 3×3 grid of immovable blocks of height 5.



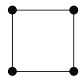
In the case where there is a hole along the upper left most vertices then each following missing vertex will be mapped to the grid below. This will ensure that the new start vertex is the next most upper left vertex.



Vertices will be separated from the other vertices by a line of grid cells in between their 3×3 mapping. Each of the corresponding edges will map to a 1×3 or 3×1 grid of cells depending on if the edge is vertical or horizontal. If there exists no edge between two vertices there will be a line of blocks of height 5 in between them.

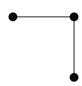


Thus, the following 2×2 grid graph would map to the corresponding Q-Billion level.



5	2	5	5	5	2	5
2	3	2	$\boxed{0}$	2	3	2
5	2	5	5	5	2	5
5	$\boxed{0}$	5	5	5	$\boxed{0}$	5
5	2	5	5	5	2	5
2	3	2	$\boxed{0}$	2	3	2
5	2	5	5	5	2	5

Note that in the mapping, the center cell highlighted in yellow does not correspond to a vertex or edge. Thus, all grid cells indexes at $(4x, 4y)\,\forall x, y \in \mathbb{Z}_{\geq 0}$ are mapped to a stack of immovable blocks of height 5. Below represents a 2×2 grid subgraph mapping.



5	2	5	5	5	2	5
2	3	2	$\boxed{0}$	2	3	2
5	2	5	5	5	2	5
5	5	5	5	5	$\boxed{0}$	5
5	5	5	5	5	2	5
5	5	5	5	2	3	2
5	5	5	5	5	$\boxed{0}$	5

The following 1×2 grid graph would map to the following Q-Billion level.

5	2	5	5	5	2	5
2	3	2	$\boxed{0}$	2	3	$\boxed{0}$
5	2	5	5	5	2	5

In general, a grid subgraph of length and height n, m would be mapped to a $3*n+n-1 \times 3*m+m-1$ grid. We will denote the resulting map of by a function $f : G[P] \mapsto \text{Level}_{\text{Q-Billion}}$. Note that the following reduction is polynomial because it maps the input subgraph to a greater constant multiple sized graph. In order to verify the following reduction is correct we must show that HPGRID'$(G[P], s)$ = True if and only if Q-Billion$(f(G[P]))$ = True. Recall that the start vertex s in HPGRID' corresponds to the upper left most vertex. Since the mouse in Q-Billion spawns in the upper leftmost cell the reduction assures they spawn in the upper left vertex.

Direction 1: HPGRID'$(G[P], s)$ = True $\implies$ Q-Billion$(f(G[P]))$ = True. If HPGRID'$(G[P], s)$ = True we know that there exists a Hamilton path starting from the upper leftmost vertex in $G[P]$. First, assume the case where every vertex in $G[P]$ has valency at least 2. Since the mouse character spawns in the top left corner grid of the Q-Billion level by design they will initially sit on a stack of immovable blocks of height 5. Either this stack will correspond to s or this will be a hole vertex that forces the mouse to move right in order to get to s. Once the mouse is at the top left grid cell of s since they can only traverse down or to the right they will drop into the vertex and be unable to climb back up to the height 5 stacks. Also, since the only way of exiting the vertex is through the allowable edges and each allowable edge has a divot of depth 2 the mouse cannot make this traversal without getting stuck. Thus, the only way to traverse an edge is if the mouse pushes the single movable box resting in the center of the vertex into the divot of depth 2. Note that the mouse can climb on top of this movable block and drop down in any of the 4 cardinal directions allowing them to successfully push the movable block in any direction to an edge. Note that if the mouse character were to push the movable box into a corner of height 5 immovable blocks, the box wold be trapped preventing the completion of the level. Thus, the mouse character must be very selective with their choices. Since each vertex has a corresponding movable box and in order for the mouse to complete the Q-Billion level they must push these movable blocks to the ground level, the mouse must visit each vertex and push each of these movable blocks to the ground level. Since the mouse character is restricted only to traversing the vertices that are connected by edges if there is a Hamilton Path in, $G[P]$ the user can successfully use the movable blocks at each vertex mapping to visit an adjacent vertex until they have reached all vertices. Once the mouse has reached the last vertex of the Hamilton path they have a single box remaining to push to the ground level. However, since in this case every vertex has valency at least 2 this implies there exists an edge that has a ground level gap that the mouse can push the remaining box into to complete the Q-Billion level.

Now consider the case where not every vertex in $G[P]$ has valency at least 2. This implies there exists at least one vertex of valency 1. Note that if there exists more than on vertex of valency 1, other than s, then there exists no Hamilton Path along $G[P]$, thus, we can ignore this case. Now consider the case where there exists one vertex of valency 1 other than s. Since this vertex is a leaf we know if there exists a Hamilton Path on $G[P]$ starting at s then it must end at the following leaf. Thus, since the mouse can successfully reach this last vertex being left with the single movable box. However, since the valency 1 vertex maps to a 3×3 grid that has it's own ground block gap in it, the mouse can successfully push this final block completing the Q-bIllion level. Thus, a True instance of HPGRID'$(G[P], s)$ maps to True instance of Q-Billion$(f(G[P]))$.

Direction 2: Q-Billion$(f(G[P]))$ = True $\implies$ HPGRID'$(G[P], s)$ = True. If Q-Billion$(f(G[P]))$ = True we know that a level resulting in a mapping from some grid graph $G[P]$ can be solved (i.e all movable blocks can be pushed down to a ground level of height 0). This requires the mouse to create a bridge that spans all the vertices by pushing a movable block into an edge. Since the mouse can visit every single vertex by making a choice of a single edge from vertex to vertex the following sub grid graph corresponding to the level also has a sequence of edges that visits every vertex. Thus, HPGRID'$(G[P], s)$ = True.

Therefore, HPGRID'$(G[P], s)$ = True if and only if Q-Billion$(f(G[P]))$ = True and Q-Billion is at least NP-Hard. □

2.3 Membership in NP

Unfortunately we were unable to directly show that every Q-Billion game must admit a polynomial certificate, which is crucial to prove that Q-Billion games itself is in NP. Specifically, we encountered the problem that in a valid solution to a game, one specific block might be moved back and forth, leaving and returning to some position several times during the process of solving a level. This could potentially lead to an exponential number of moves, a similar scenario to the game Sokoban. Nevertheless, our research has concluded in the following results that could potentially lead to a full proof of NP membership.

Theorem 2.4. *There cannot exists a solvable Q-Billion level with a 2×2 grid of blocks of height greater than 1.*

Proof. Any step that reduces the total height of the stacks in a restricted 2×2 grid must push the blocks out of the region since the mouse cannot push two block together. We can continue pushing blocks out of this region until all of the blocks are of the same height. Thus, if the height is ever greater than 1 the level will be unsolvable. In other words if we start with a 2×2 region of stacks with a minimum height x, we will at best end up with 2×2 of height x. When $x \geq 2$, the level will be unsolvable. □

Since we added the constraint that the mouse cannot go up by more than 1 block in each step, there must be a "staircase" (i.e. a connected gadget of height $n-1$, $n-2$, ..., 2, 1) attached to each stack $n \geq 2$ for that level otherwise, one would not able to reach the height of the top stack. In addition, the way that the player moves the blocks around the "staircase" is limited: if one pushed two adjacent blocks on a staircase towards the same direction, it might lead to a 2×2 region of blocks with height greater than 1 and thus render an unsolvable level. In such a way, we believe that the number of potential ways to complete a solvable game is relatively limited.

In another perspective, we consider the maximum number of movable blocks over the whole grid. It is obvious that the total number of heights over each stack must be less than or equal to the number of grids on the board. Utilizing this constraint, we arrive at the following theorem:

Theorem 2.5. *The maximum height of blocks over a $n \times n$ board is at most $O(n^{\frac{2}{3}})$.*

Proof. If there is a block with height k, there must be an associated "staircase" $1, 2, ..., k-1$, summing $S_k = k(k+1)/2$ combined with the height k block. In addition, for the block with height $k-1$, there also have to be an associated "staircase", summing $S_{k-1} = (k-1)k/2$. So we must have the sum of heights be at least $S_1 + ... + S_k = \frac{\sum_1^n k(k+1)}{2}$. Using the transform $k(k+1) = (k(k+1)(k+2) - (k-1)(k)(k+1))/3$, the above sum is equal to $k(k+1)(k+2)/6$, which must be less than or equal to n^2. So k is at most $O(n^{\frac{2}{3}})$. □

From the above analysis and theorem, we speculate that the restricted version of Q-Billion is in NP due to the limited nature of a solvable game (both in highest block and in shape of block chunks) and also the patterns of a potentially valid certificate (any steps must not create a 2×2 region with all blocks greater than 1).

3 Future Directions

One potential future direction is to find a numerical upper bound of heights of blocks in a solvable game. It is notable that creating games with maximum height 4 or 5 is possible. However, we are not able to create a valid game for maximum height 6. Considering this, any proof of an upper bound on heights of blocks could play a vital part in adding limitations to a game process and the credential. Moreover, we may incorporate the idea of "gadgets": find some simple combinations of blocks with certain height, where such combination can be easily proved as admitting polynomial credential, and then divide the whole game into several disjoint gadgets. But a clear challenge to this method is to partition the game into disjoint parts such that no part affect others effectively. Overall, we had a great time researching and trying to identify the complexity class of Q-Billion. Thank you.

References

[1] *Q-Billion Image.*

[2] *Game Boy World Direct Footage 022: Q Billion.* Dec 2014.

[3] Alon Itai, Christos H Papadimitriou, and Jayme Luiz Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, 1982.