### Algorithm: Hamiltonian Path

Princess Elara needed to find a path through all the islands without revisiting any, to ensure the safety of her fleet. She used the Hamiltonian Path algorithm to find such a path.

#### Initialize Data Structures:

* Princess Elara used a magical map (list) to keep track of the path.

#### Find Hamiltonian Path:

* She recursively explored each island, checking if it could be added to the path.

#### Implementation:

| **def** is\_valid(v, pos, path, islands):  **if** islands[path[pos - 1]][v] == 0:  **return** **False**  **for** vertex **in** path:  **if** vertex == v:  **return** **False**  **return** **True**  **def** hamiltonian\_path\_util(islands, path, pos):  **if** pos == len(islands):  **return** **True**  **for** v **in**  range(1, len(islands)):  **if** is\_valid(v, pos, path, islands):  path[pos] = v  **if** hamiltonian\_path\_util(islands, path, pos + 1):  **return** **True**  path[pos] = -1  **return** **False**  **def** hamiltonian\_path(islands):  path = [-1] \* len(islands)  path[0] = 0  **if** **not** hamiltonian\_path\_util(islands, path, 1):  **return** []  **return** path  *# Example usage:*  islands = [  [0, 1, 0, 1, 0],  [1, 0, 1, 1, 1],  [0, 1, 0, 0, 1],  [1, 1, 0, 0, 1],  [0, 1, 1, 1, 0]  ]  print(hamiltonian\_path(islands)) *# Output: [0, 1, 2, 4, 3] or any valid path* |
| --- |

#### Explanation:

Initialize:

* path: A magical map to track the Hamiltonian path.

Find Hamiltonian Path:

### Princess Elara recursively explored each island, checking if it could be added to the path.