- Backtrack(V, n)
    - If reject(V, n) = true
        - Return false
    - If accept(V, n) = true
        - Return true
    - Else
        - V = firstMove(V, n)
        - While validMoveExists(V, n)
            - If Backtrack(V, n) = true
                - Return true
        - Return false

- Reject(V, n)
    - If validMoveExists(V, n) = false AND 1 is in V
        - Return true
    - Else return false

- Accept(V, n)
    - If 1 is not in V
        - Return true

- firstMove(V, n)
    - //Jump the first possible coin, making its position 0 and the destination +1
    - //remove resultant 0 from list
    - //Increment n (passed by reference)
    - //return result

- validMoveExists(V, n)
    - //iterate through list
    - //if an element =1 is adjacent to 2 and 2 is adjacent to another 1
        - Return true
    - If an element =1 is adjacent to 1 and the second 1 is adjacent to another 1
        - Return true
    - Else return false