```java
public static boolean canWin(int[] arr, int leap) {
        boolean answer = false;
        int stack = 0;
        boolean result = move(0, -1, arr, leap, answer, stack);
        return result;
    }

    public static boolean move(int currIndex, int prevIndex, int[] arr, int leap, boolean answer, int stack) {
        // base case START---------------------------
        if ((currIndex + leap) >= arr.length) {
            System.out.println("yes possible");
            answer = true;
            return true;
        }
        if ((currIndex + 1) >= arr.length) {
            answer = true;
            return true;
        }
        // base case END---------------------------

        // step backwards
        if ((currIndex - 1) >= 0) {
            if ((arr[currIndex - 1] == 0) && (currIndex - 1) != prevIndex && (currIndex - 1) > stack) {
                System.out.println("step backwards once: curr: " + currIndex + " prev: " + prevIndex);
                move((currIndex - 1), currIndex, arr, leap, answer, stack);
            }
        }

        // leap backwards
        if ((currIndex - leap) >= 0) {
            if ((arr[currIndex - leap] == 0) && (currIndex - leap) != prevIndex && (currIndex - leap) > stack) {
                System.out.println("step backwards LEAP: curr: " + currIndex + " prev: " + prevIndex);
                move((currIndex - leap), currIndex, arr, leap, answer, stack);
            }
        }

        // Update Stack
        // updating it here ensures we have traversed backwards as much as possible
        stack = currIndex;
        System.out.println("Stack updation reached: stackVal: " + stack);

        // step forward
        if ((currIndex + 1) < arr.length) {
            if ((arr[currIndex + 1] == 0) && (currIndex + 1) != prevIndex && (currIndex + 1) > stack) {
                System.out.println("step forward once: curr: " + currIndex + " prev: " + prevIndex);
                move((currIndex + 1), currIndex, arr, leap, answer, stack);
            }
        }

        // leap forward
        if ((currIndex + leap) < arr.length) {
            if ((arr[currIndex + leap] == 0) && (currIndex + leap) != prevIndex && (currIndex + leap) > stack) {
                System.out.println("step forward LEAP: curr: " + currIndex + " prev: " + prevIndex);
                move((currIndex + leap), currIndex, arr, leap, answer, stack);
            }
        }
        return answer;
    }
```