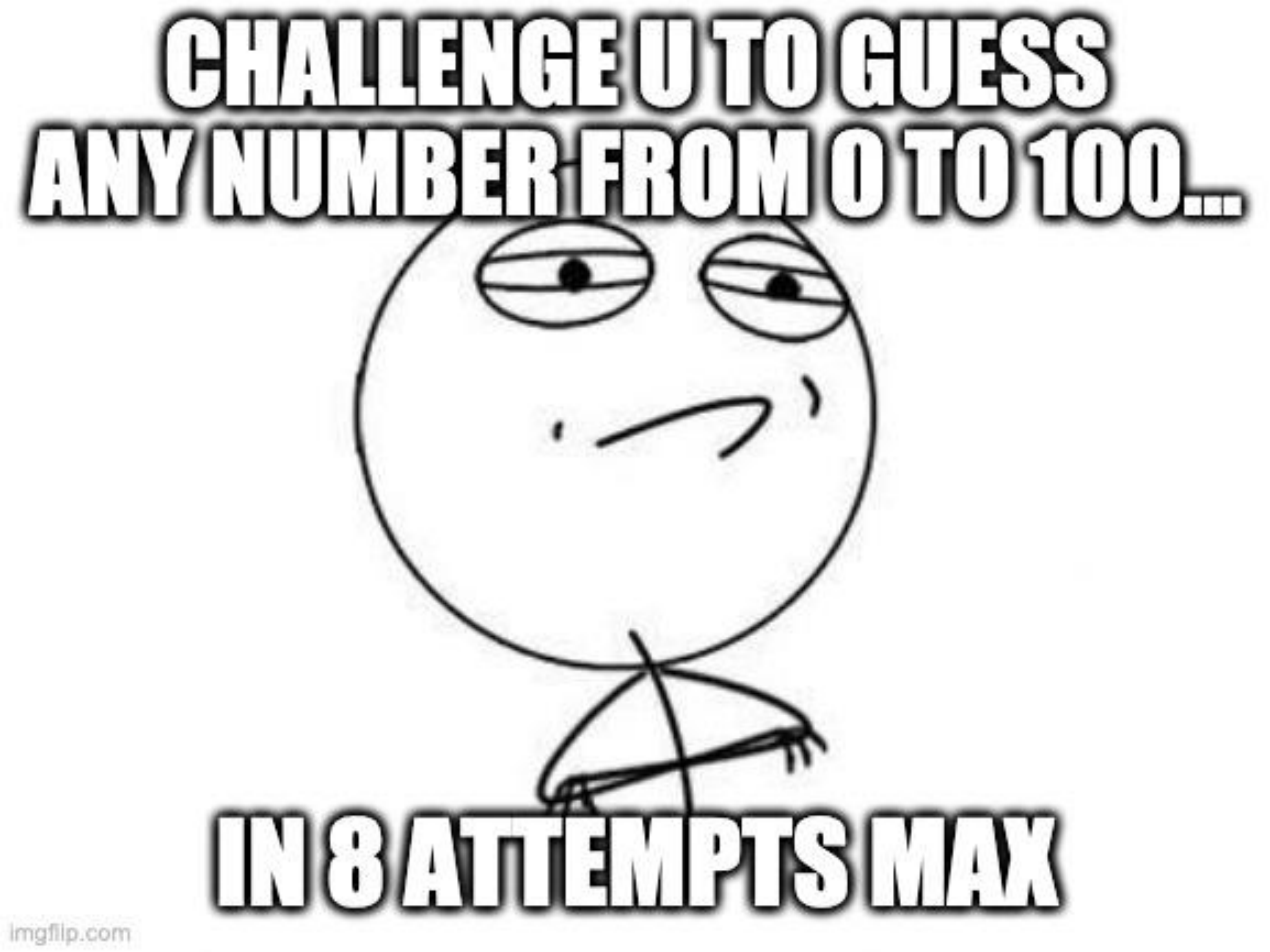
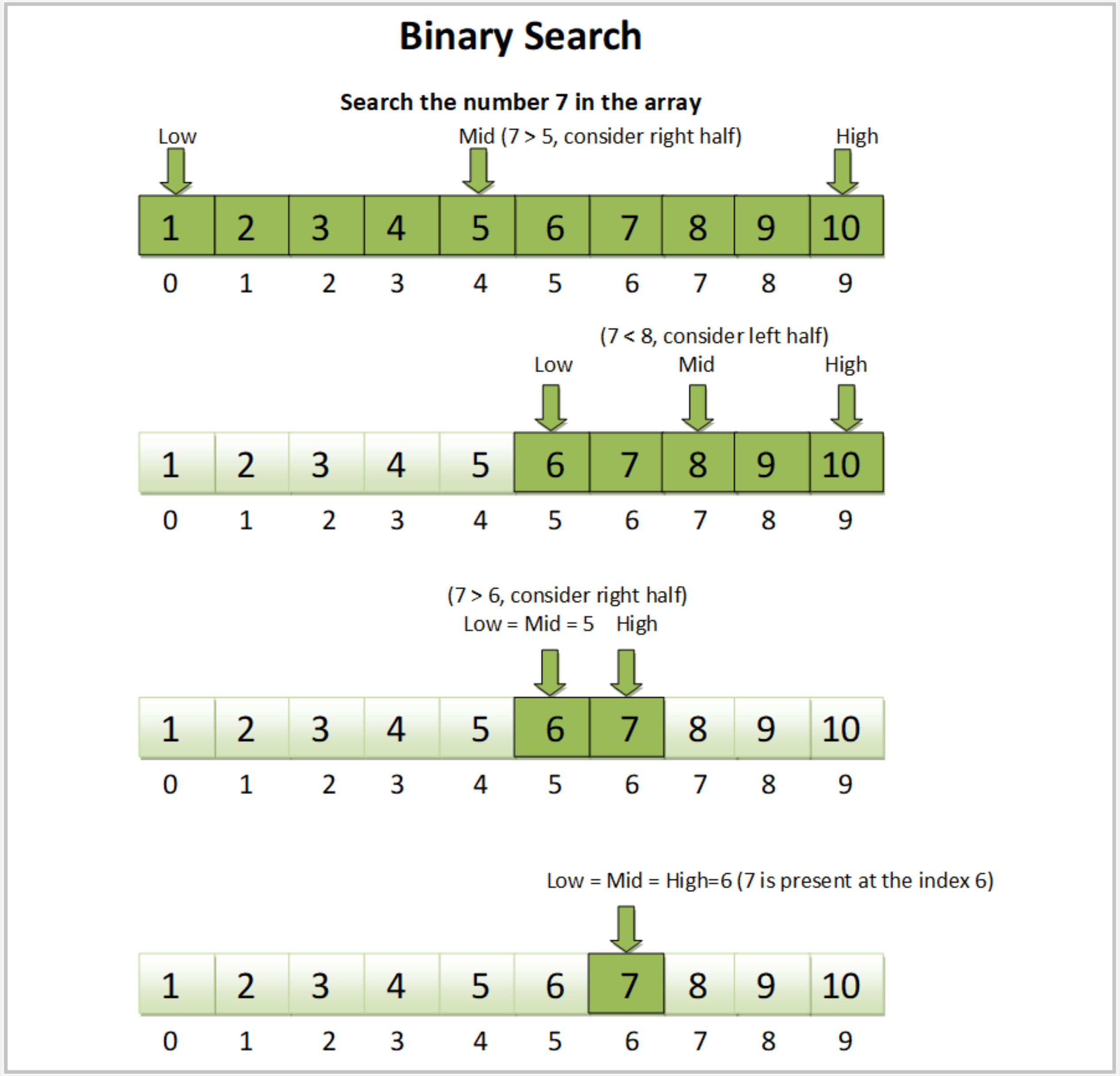
BINARY SEARCH


CHALLENGE U TO GUESS
ANY NUMBER FROM 0 TO 100...
IN 8 ATTEMPTS MAX
imgflip.com

AND WHAT ABOUT FROM 0 TO 1000...
IN 11 ATTEMPTS MAX???
imgflip.com

Do I need to buy apples?



Binary Search

Search the number 7 in the array

Low

Mid (7 > 5, consider right half)

High

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

(7 < 8, consider left half)

Low

Mid

High

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

(7 > 6, consider right half)

Low = Mid = 5 High

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

Low = Mid = High=6 (7 is present at the index 6)

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

https://www.cs.usfca.edu/~galles/visualization/Search.html

GET TO THE CODE
DO IT, DO IT NOW!!
imgflip.com

Cool! But…
How efficient is this?

HOW MANY TIMES DO WE SPLIT RANGE IN HALF UNTIL WE GET JUST ONE NUMBER?

$$1 = N / 2^x$$

$$2^x = N$$

$$lg_2(2^x) = lg_2 (N)$$

$$x * lg_2(2) = lg_2 (N)$$

$$x * 1 = lg_2 (N)$$

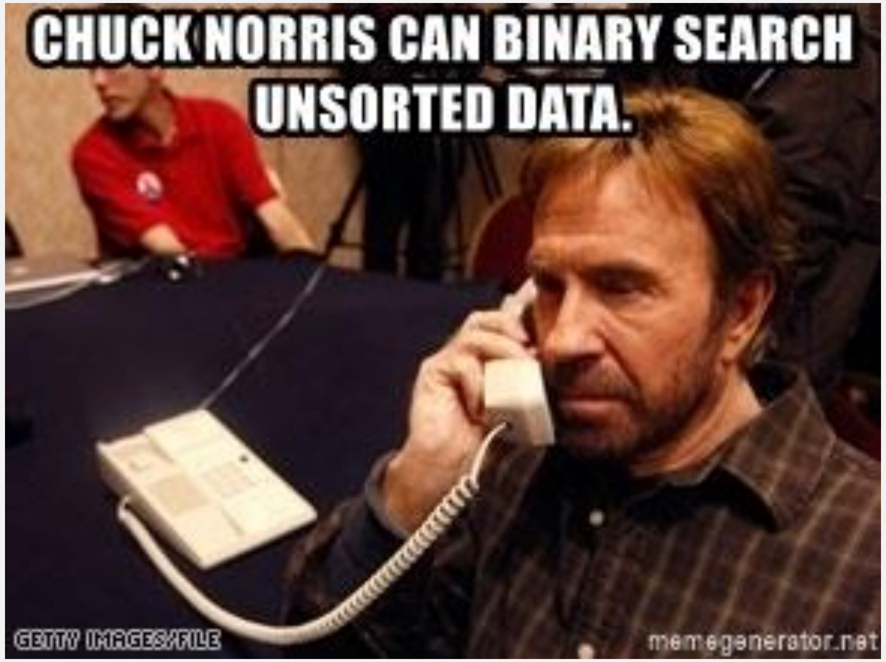
$$x = lg_2 (N)$$

O(LOG(N))

REVIEW

CAN BINARY SEARCH WORK WITH AN UNORDERED INPUT?

- a) Yes
- b) No
- c) Hell, no



HOW WOULD BINARY SEARCH PERFORM ON A LINKED LIST?

- a) O(1)

- b) O(log N)

- [✓ c)] O(N)   You need to traverse the entire list to find the mid point.

EXERCISE

- Code a new version of the **`binary_search`** function that can find elements in a circularly sorted array (for instance `[8, 9, 10, 2, 5, 6]`).

EXERCISE

- Code a function **`binary_search_count`** that calculates, for a given sorted ***array*** (ascending order), and a given number ***n***, the number of occurrences of that number ***n*** within the array.