Number Hunt — Results and Graph

Name: _____ **Date:** _____

Part 1: Binary Search Rules

Always guess the MIDPOINT of the remaining range.

```
midpoint = (lowest_possible + highest_possible) ÷ 2 (round down)
```

Example trace — number is 73, range starts 1–100:

Step	Range	Midpoint guess	Result
1	1–100	50	HIGHER
2	51–100	75	LOWER
3	51–74	62	HIGHER
4	63–74	68	HIGHER
5	69–74	71	HIGHER
6	72–74	73	CORRECT!

6 guesses to find 73 out of 100 numbers.

Part 2: Practice

Number is 31. Complete the binary search:

Step	Range	Midpoint guess	Higher/Lower/Correct
1	1–100	50	
2			
3			
4			
5			
6			

Part 3: Game Results

Partner's number was found in ___ guesses each round:

Round	Number chosen	Guesses needed

Round	Number chosen	Guesses needed
1		
2		
3		
4		
5		
Average		

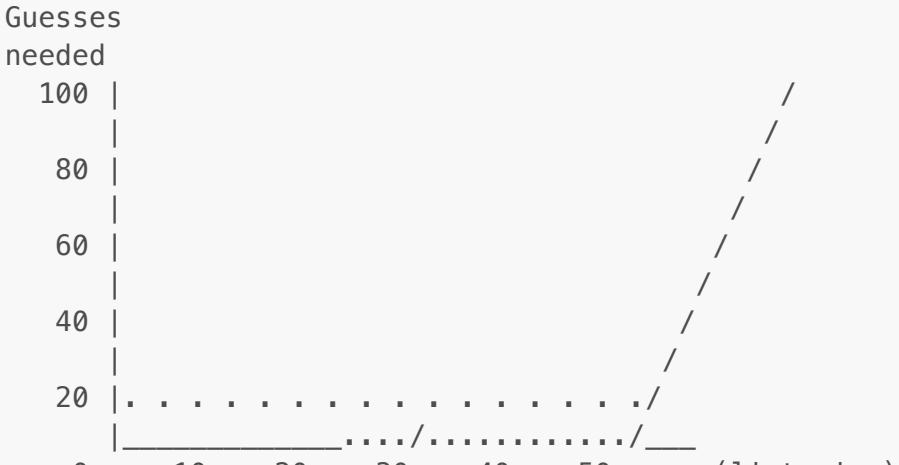
Part 4: Comparison Table

List size (n)	Linear search worst case	Binary search worst case
10	10	
100	100	
1,000	1,000	
10,000	10,000	
1,000,000	1,000,000	≈ 20

Hint for binary search: count how many times you can halve n before reaching 1. That's $\log_2(n)$.

Part 5: Graph

Plot both curves on these axes. Note: linear search shoots off the chart — mark this!



Binary search (....) stays nearly flat. Linear search (/) grows steeply.

What shape is the binary search curve? _____

Part 6: Reflection

Why MUST the list be sorted for binary search to work?

Can you think of a real situation where you use binary search without knowing it?

If you had to search 1 million sorted records, which algorithm would you choose? How much faster is it?
