

# Complexity Olympics — Results Recording Sheet

---

Name: \_\_\_\_\_ Date: \_\_\_\_\_

---

## Station 1: $O(1)$ — Constant Time

Input size (n)	Steps taken
10	
100	
1,000	

Pattern I noticed: \_\_\_\_\_

---

## Station 2: $O(\log n)$ — Logarithmic Time

Input size (n)	Guesses needed
8	
16	
32	

Pattern I noticed: \_\_\_\_\_

Predict:  $n=64$  would need \_\_\_ guesses.  $n=1024$  would need \_\_\_ guesses.

---

## Station 3: $O(n)$ — Linear Time

Input size (n)	Cards examined
5	
10	
20	

Pattern I noticed: \_\_\_\_\_

---

## Station 4: $O(n^2)$ — Quadratic Time

Input size (n)	Actual comparisons	Formula: $n \times (n-1) \div 2$
3		

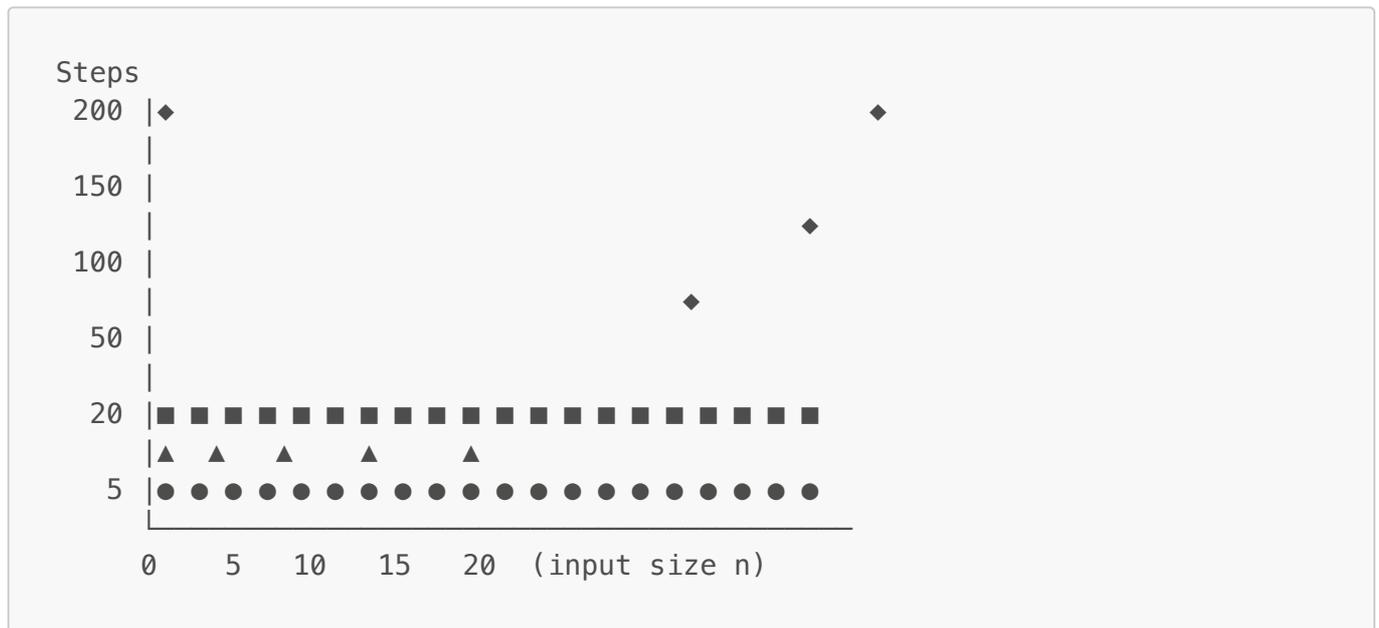
---

Input size (n)	Actual comparisons	Formula: $n \times (n-1) \div 2$
4		
5		
6		

Pattern I noticed: \_\_\_\_\_

## Graph — All Four Complexity Classes

Plot on the axes below. Use different symbols for each:  $O(1)=\bullet$ ,  $O(\log n)=\blacktriangle$ ,  $O(n)=\blacksquare$ ,  $O(n^2)=\blacklozenge$



Sketch your actual data on the graph above.

## Predictions for n=100

Complexity	Formula	Steps for n=100
$O(1)$	always 1	1
$O(\log n)$	$\log_2(100) \approx 7$	7
$O(n)$	n	100
$O(n^2)$	$n \times (n-1) \div 2$	

## Real-World Decisions

A social network has 1 billion users. A search through all users to find a match takes  $O(n)$  time. How many steps would that take? Why would this be unacceptable?

If you could choose between  $O(n)$  and  $O(n^2)$  for sorting 10,000 names, which would you choose? How much better is it?

---

---

## Big-O Summary

Fill in the final column:

Name	Notation	Growth rate	Real-world example
Constant	$O(1)$	Always the same	
Logarithmic	$O(\log n)$	Grows very slowly	
Linear	$O(n)$	Proportional to $n$	
Quadratic	$O(n^2)$	Squares with $n$	