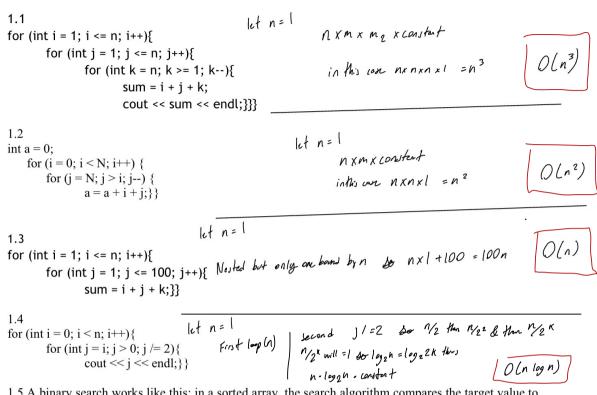
## Question 1. What is the time complexity for the following code/program?



1.5 A binary search works like this: in a sorted array, the search algorithm compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value and repeating this until the target value is found. If the search ends with the remaining half being empty, the target is not in the array. What is the complexity of binary search? Why?

Jumediation the best case is I, where we succeed on our first "great".

Our first array would be no than 2nd would be note making it 2nd which like 1.4 will become I be thus log\_n = log\_2 2 k giving us the worst out come of k = log\_2(n) become I be thus log\_n = log\_2 all the way down the entirity of the last or the volve is actually missing brows and list.

## Question 2. Finish the following table.

Expression	Dominant term(s)	$O(\ldots)$
$5 + 0.001n^3 + 0.025n$	0.001 n3	N3
$500n + 100n^{1.5} + 50n \log_{10} n$	100n'.5	n 1.5
$0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$	2.5·n 1.75	n 1.75
$n^2 \log_2 n + n(\log_2 n)^2$	nelogin	n²log n
$n \log_3 n + n \log_2 n$	n logan	nlog n
$3\log_8 n + \log_2 \log_2 \log_2 n$	31098 n	leg n
$100n + 0.01n^2$	0.01 n <sup>2</sup>	n <sup>2</sup>
$0.01n + 100n^2$	100n2	n <sup>2</sup>
$2n + n^{0.5} + 0.5n^{1.25}$	0.5n 1.25	1.25 N
$0.01n\log_2 n + n(\log_2 n)^2$	n[109,n)2	ار وهاام
$100n \log_3 n + n^3 + 100n$	n³	n³
$0.003\log_4 n + \log_2\log_2 n$	log2 (log2 n)	(n eol) gol