

# Binary Search Recursion

															high	
															mid	
A	3	6	8	12	14	17	26	29	31	36	42	47	53	55	62	
low	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
l - low			key - what you want to find												h - high	

Algorithm BinarySearch(l, h, key) —  $T(n)$

{  
if (l == h) { if (A[l] == key) return l; } — }  
else return 0;  
} — prevents overflow

} else

$$\text{mid} = \text{Low} + \frac{(\text{high} - \text{low})}{2}$$

if (key == A[mid]) return mid; — 1

if (key < A[mid]) { — 1

return BinarySearch(l, mid - 1, key);

}

else

return BinarySearch(mid + 1, h, key);

}

}

$O(\log n)$

$$T(n) = \begin{cases} 1 & n \leq 1 \\ T(n/2) & n > 1 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

$$\log_2 n = 0 \text{ and } O(\log n)$$

# Iterative

							mid							high	
A	3	6	8	12	14	17	26	29	31	36	42	47	53	59	62
low	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	↑														↑

$k = 42$        $low, high, mid = \left\lfloor \frac{low + high}{2} \right\rfloor$

$11 \quad 11 \quad \frac{11+11}{2} = 11$

$1 \quad 19 \quad \frac{1+19}{2} = 8$

$9 \quad 19 \quad \frac{9+19}{2} = 12$

$9 \quad 11 \quad \frac{9+11}{2} = 10$

$low \quad mid \quad mid \quad mid$

							mid								high
A	3	6	8	12	14	17	26	29	31	36	42	47	53	59	62
low	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		↓									↑				
											high				

A	3	6	8	12	14	17	26	29	31	36	42	47	52	56	62
	low	1	2	3	4	5	6	7	8	9	10	11	12	13	14

low = 1  
high = 15  
mid = 8  
key = 12



low	high	$mid = \left\lfloor \frac{low + high}{2} \right\rfloor$
1	15	8
1	7	4



int BinSearch(A, n, key)

low = 1

high = n

while (low ≤ high)

{

mid = low +  $\left(\frac{\text{high} - \text{low}}{2}\right)$

if (key == A[mid])

return mid;

if (key < A[mid])

high = mid - 1;

else

low = mid + 1;

}

return 0;