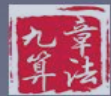


Binary Search & Rotated Sorted Array

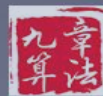
课程尚未开始, 请耐心等待



关注九章算法微信
获得最新面试题、
题解、面经分享



Binary Search



Classical Binary Search

Given an sorted integer array - nums, and an integer - target. Find the any/first/last position of target in nums, return -1 if target doesn't exist.

```
public int binarySearch(int[] nums, int target)
```



How we do binary search?

	start				end				
	↓				↓				
index	0	1	2	3	4	5	6	7	8
nums	2	3	5	8	13	21	34	55	89

1. Find 5

How we do binary search?

	start						mid		end	
	↓						↓		↓	
index	0	1	2	3	4	5	6	7	8	
nums	2	3	5	8	13	21	34	55	89	

1. Find 5, mid=4

How we do binary search?

	start		mid		end				
	↓		↓		↓				
index	0	1	2	3	4	5	6	7	8
nums	2	3	5	8	13	21	34	55	89

1. Find 5, mid=4, 2. Find it!

How we do binary search?

			start	mid	end				
			↓	↓	↓				
index	0	1	2	3	4	5	6	7	8
nums	2	3	5	8	13	21	34	55	89

1. Find 5, mid=4, 2. Find it!

2. Find 8, mid=4, 2, 3. Find it!



How we do binary search?

	start				mid				end
	↓				↓				↓
index	0	1	2	3	4	5	6	7	8
nums	2	3	5	8	13	21	34	55	89

1. Find 5, mid=4, 2. Find it!
2. Find 8, mid=4, 2, 3. Find it!
3. Find 14, mid=4



How we do binary search?

					start		mid		end
					↓		↓		↓
index	0	1	2	3	4	5	6	7	8
nums	2	3	5	8	13	21	34	55	89

1. Find 5, mid=4, 2. Find it!
2. Find 8, mid=4, 2, 3. Find it!
3. Find 14, mid=4, 6



How we do binary search?

					start	mid	end		
					↓	↓	↓		
index	0	1	2	3	4	5	6	7	8
nums	2	3	5	8	13	21	34	55	89

1. Find 5, mid=4, 2. Find it!
2. Find 8, mid=4, 2, 3. Find it!
3. Find 14, mid=4, 6, 5



How we do binary search?

start, mid
end

index	0	1	2	3	4	5	6	7	8
nums	2	3	5	8	13	21	34	55	89

1. Find 5, mid=4, 2. Find it!
2. Find 8, mid=4, 2, 3. Find it!
3. Find 14, mid=4, 6, 5, 4. Return -1



复杂度分析

$$T(n) = T(n/2) + O(1) = O(\log n)$$



六脉神剑之“少商剑”

算法面试中如果需要优化 $O(n)$ 的时间复杂度
那么只能是 $O(\log n)$ 的二分法



Recursion or While-Loop?

Binary Search



程序实现中的常见问题

- 又死循环了！
 - what are you 弄撒捏！
- 循环结束条件到底是哪个？
 - $\text{start} \leq \text{end}$
 - $\text{start} < \text{end}$
 - $\text{start} + 1 < \text{end}$
- 指针变化到底是哪个？
 - $\text{start} = \text{mid}$
 - $\text{start} = \text{mid} + 1$
 - $\text{start} = \text{mid} - 1$



通用的Binary Search模板

<http://www.lintcode.com/en/problem/binary-search/>

<http://www.ninechapter.com/solutions/binary-search/>

四要素:

1. $start + 1 < end$
2. $start + (end - start) / 2$
3. $A[mid] ==, <, >$
4. $A[start] A[end] ? target$



Find First Position VS Find Last Position



Search for a range

<http://www.lintcode.com/problem/search-for-a-range/>

<http://www.jiuzhang.com/solutions/search-for-a-range/>



Search Insert Position

<http://www.lintcode.com/problem/search-insert-position/>

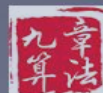
<http://www.jiuzhang.com/solutions/search-insert-position/>



Search a 2D Matrix

<http://www.lintcode.com/problem/search-a-2d-matrix/>

<http://www.jiuzhang.com/solutions/search-a-2d-matrix/>



Search a 2D Matrix II

<http://www.lintcode.com/problem/search-a-2d-matrix-ii/>

[0, 1, 2, 4]
[1, 2, 6, 9]
[3, 5, 7,10]
[7, 8, 9,11]



Search in a 2D Matrix

28	29	31	36	36	43	49	54	58	64	66	68	68	77	84	89	97	97	97	105
33	38	39	48	53	58	61	62	62	69	70	71	71	77	90	91	103	105	108	110
33	44	51	56	62	63	67	69	74	83	86	91	96	98	104	110	110	110	118	125
41	44	52	59	71	75	83	92	100	107	114	122	122	124	132	133	140	141	145	146
43	49	54	59	72	79	83	101	104	115	122	124	124	124	140	140	140	142	151	154
45	52	55	62	72	80	92	104	111	122	130	132	132	136	149	156	160	165	171	179
51	60	68	73	81	86	94	110	115	129	138	141	146	149	157	163	169	178	186	191
51	61	74	83	92	95	95	111	120	129	138	143	146	155	159	171	180	188	196	201
56	67	81	89	94	98	98	119	123	133	147	149	149	164	167	174	183	188	205	211
62	69	85	95	97	101	105	119	131	139	152	152	160	165	171	178	190	195	212	215
71	72	90	95	106	111	116	121	131	144	155	160	163	173	180	186	197	199	220	221
79	87	99	104	108	115	123	130	140	144	161	166	174	177	189	192	205	209	225	230
84	94	103	106	117	120	129	133	145	153	165	173	175	178	196	199	209	209	231	235
91	96	105	108	120	128	137	141	151	154	169	182	187	192	201	209	214	220	233	237
95	100	109	110	123	137	139	141	159	161	174	184	188	201	205	213	218	228	233	241
101	107	115	119	130	146	155	155	168	173	178	187	190	202	209	221	226	230	237	249
109	109	115	125	131	148	156	164	173	180	180	196	204	212	217	222	232	240	249	253
113	117	124	126	138	151	157	167	181	183	184	204	213	219	223	231	236	242	250	253
115	123	131	138	142	152	160	167	184	190	197	210	215	224	225	238	244	246	256	258
118	126	131	138	149	161	170	176	184	193	206	217	217	225	234	240	249	257	265	267

Search in a 2D Matrix

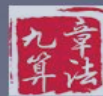
28	29	31	36	36	43	49	54	58	64	66	68	68	77	84	89	97	97	97	105
33	38	39	48	53	58	61	62	62	69	70	71	71	77	90	91	103	105	108	110
33	44	51	56	62	63	67	69	74	83	86	91	96	98	104	110	110	110	118	125
41	44	52	59	71	75	83	92	100	107	114	122	122	124	132	133	140	141	145	146
43	49	54	59	72	79	83	101	104	115	122	124	124	124	140	140	140	142	151	154
45	52	55	62	72	80	92	104	111	122	130	132	132	136	149	156	160	165	171	179
51	60	68	73	81	86	94	110	115	129	138	141	146	149	157	163	169	178	186	191
51	61	74	83	92	95	95	111	120	129	138	143	146	155	159	171	180	188	196	201
56	67	81	89	94	98	98	119	123	133	147	149	149	164	167	174	183	188	205	211
62	69	85	95	97	101	105	119	131	139	152	152	160	165	171	178	190	195	212	215
71	72	90	95	106	111	116	121	131	144	155	160	163	173	180	186	197	199	220	221
79	87	99	104	108	115	123	130	140	144	161	166	174	177	189	192	205	209	225	230
84	94	103	106	117	120	129	133	145	153	165	173	175	178	196	199	209	209	231	235
91	96	105	108	120	128	137	141	151	154	169	182	187	192	201	209	214	220	233	237
95	100	109	110	123	137	139	141	159	161	174	184	188	201	205	213	218	228	233	241
101	107	115	119	130	146	155	155	168	173	178	187	190	202	209	221	226	230	237	249
109	109	115	125	131	148	156	164	173	180	180	196	204	212	217	222	232	240	249	253
113	117	124	126	138	151	157	167	181	183	184	204	213	219	223	231	236	242	250	253
115	123	131	138	142	152	160	167	184	190	197	210	215	224	225	238	244	246	256	258
118	126	131	138	149	161	170	176	184	193	206	217	217	225	234	240	249	257	265	267

Search in a 2D Matrix

28	29	31	36	36	43	49	54	58	64	66	68	68	77	84	89	97	97	97	105
33	38	39	48	53	58	61	62	62	69	70	71	71	77	90	91	103	105	108	110
33	44	51	56	62	63	67	69	74	83	86	91	96	98	104	110	110	110	118	125
41	44	52	59	71	75	83	92	100	107	114	122	122	124	132	133	140	141	145	146
43	49	54	59	72	79	83	101	104	115	122	124	124	124	140	140	140	142	151	154
45	52	55	62	72	80	92	104	111	122	130	132	132	136	149	156	160	165	171	179
51	60	68	73	81	86	94	110	115	129	138	141	146	149	157	163	169	178	186	191
51	61	74	83	92	95	95	111	120	129	138	143	146	155	159	171	180	188	196	201
56	67	81	89	94	98	98	119	123	133	147	149	149	164	167	174	183	188	205	211
62	69	85	95	97	101	105	119	131	139	152	152	160	165	171	178	190	195	212	215
71	72	90	95	106	111	116	121	131	144	155	160	163	173	180	186	197	199	220	221
79	87	99	104	108	115	123	130	140	144	161	166	174	177	189	192	205	209	225	230
84	94	103	106	117	120	129	133	145	153	165	173	175	178	196	199	209	209	231	235
91	96	105	108	120	128	137	141	151	154	169	182	187	192	201	209	214	220	233	237
95	100	109	110	123	137	139	141	159	161	174	184	188	201	205	213	218	228	233	241
101	107	115	119	130	146	155	155	168	173	178	187	190	202	209	221	226	230	237	249
109	109	115	125	131	148	156	164	173	180	180	196	204	212	217	222	232	240	249	253
113	117	124	126	138	151	157	167	181	183	184	204	213	219	223	231	236	242	250	253
115	123	131	138	142	152	160	167	184	190	197	210	215	224	225	238	244	246	256	258
118	126	131	138	149	161	170	176	184	193	206	217	217	225	234	240	249	257	265	267

First Bad Version

<http://www.lintcode.com/problem/first-bad-version/>
<http://www.jiuzhang.com/solutions/first-bad-version/>



Find Peak Element

<http://www.lintcode.com/problem/find-peak-element/>

<http://www.jiuzhang.com/solutions/find-peak-element/>



5 minutes break



Rotated Sorted Array



Find Minimum in Rotated Sorted Array

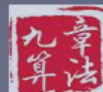
<http://www.lintcode.com/problem/find-minimum-in-rotated-sorted-array/>

<http://www.jiuzhang.com/solutions/find-minimum-in-rotated-sorted-array/>



Find Minimum in Rotated Sorted Array II

<http://www.lintcode.com/zh-cn/problem/find-minimum-in-rotated-sorted-array-ii/>



Search in Rotated Sorted Array

<http://www.lintcode.com/problem/search-in-rotated-sorted-array/>

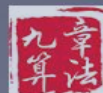
<http://www.jiuzhang.com/solutions/search-in-rotated-sorted-array/>



Search in Rotated Sorted Array II

<http://www.lintcode.com/problem/search-in-rotated-sorted-array-ii/>

Linear algorithm, go for-loop!
[1,1,1,...1] with only one 2 in it



Median of Two Sorted Arrays

<http://www.lintcode.com/problem/median-of-two-sorted-arrays/>

<http://www.ninechapter.com/solutions/median-of-two-sorted-arrays/>



Recover Rotated Sorted Array

<http://www.lintcode.com/problem/recover-rotated-sorted-array/>

<http://www.jiuzhang.com/solutions/recover-rotated-sorted-array/>



Rotate String

<http://www.lintcode.com/problem/rotate-string/>
<http://www.jiuzhang.com/solutions/rotate-string/>



Reverse Words in a String

<http://www.lintcode.com/problem/reverse-words-in-a-string/>

<http://www.jiuzhang.com/solutions/reverse-words-in-a-string/>



Conclusion

1. Binary Search Template (4 key points)
2. Rotated Sorted Array
 - a. Find Minimum
 - b. Find Target
 - c. why $O(n)$ with duplicates ?
3. Find Median in Two Sorted Array
find kth
4. Reverse in 3 steps

