

王道考研-机试指南

www.cskaoyan.com

王道考研-机试指南

第三章 排序与查找



排序与查找

排序

查找

无序 → 有序

希尔排序

快速排序

冒泡排序

基数排序

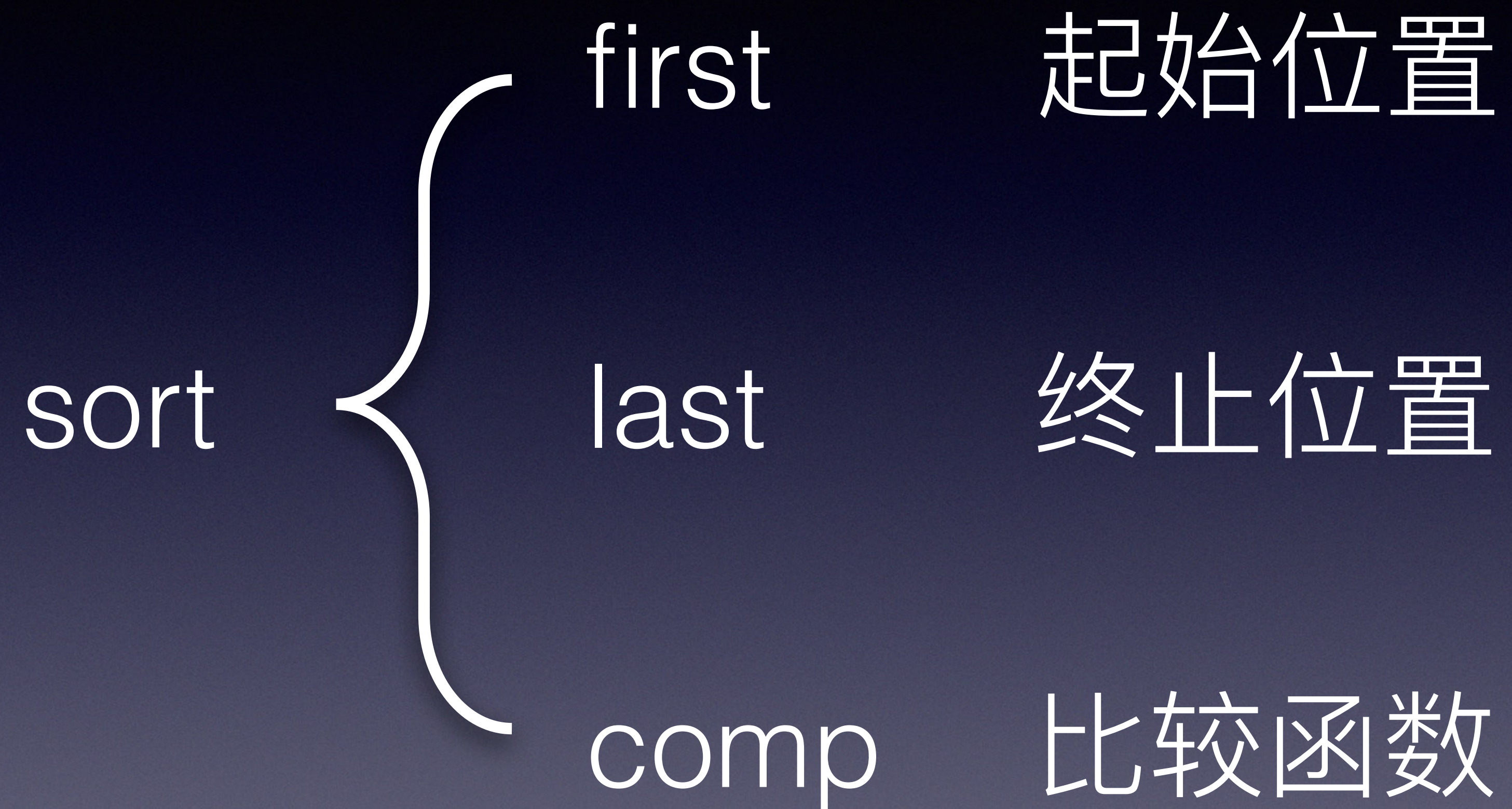
排序

归并排序

选择排序

插入排序

sort





无序 → 有序

排序

内定义数据类型

自定义数据类型

排序



内定义数据类型

自定义数据类型

例题3.1 排序

题目描述：

对输入的 n 个数进行排序并输出。

输入：

输入的第一行包括一个整数 n 。接下来的一行包括 n 个整数。

输出：

可能有多组测试数据，对于每组数据，输出排序后的 n 个整数,每个数后面都有一个空格。每组测试数据的结果占一行。

样例输入：

```
4
1 4 3 2
```

样例输出：

```
1 2 3 4
```


排序



内定义数据类型

自定义数据类型

排序

内定义数据类型

自定义数据类型

设计比较函数

定义大小关系

例题3.2 成绩排序

题目描述：

用一维数组存储学号和成绩，然后按成绩排序输出。

输入：

输入的第一行中包括一个整数N，它代表学生的个数。接下来的N行中，每行包括两个整数p和q，分别代表每个学生的学号和成绩。

输出：

按照学生的成绩从小到大进行排序，并将排序后的学生信息打印出来。如果学生的成绩相同，那么按照学号的大小从小到大排序。

样例输入：

```
3
1 90
2 87
3 92
```

样例输出：

```
2 87
1 90
3 92
```




习题3.2 整数奇偶排序

题目描述

输入10个整数，彼此以空格分隔。重新排序以后输出(也按空格分隔)，要求：
1.先输出其中的奇数,并按从大到小排列； 2.然后输出其中的偶数,并按从小到大排列。

输入：

任意排序的10个整数（0~100），彼此以空格分隔。

输出：

可能有多组测试数据，对于每组数据，按照要求排序后输出，由空格分隔。


样例输入：

4 7 3 13 11 12 0 47 34 98

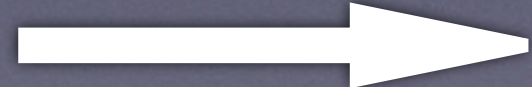
样例输出：

47 13 11 7 3 0 4 12 34 98

排序算法的特性

线性排序  计数排序

逆序数对  归并排序

第K大数  快速排序

排序算法的特性

线性排序 → 计数排序

逆序数对 → 归并排序

第K大数 → 快速排序

$O(n \log n)$

基于比较

线性排序

$O(n + r)$

给定范围 r

计数排序

5	3	2	4	3	3	2	7	7	1
---	---	---	---	---	---	---	---	---	---

0	1	2	3	1	1	0	2	0	0
0	1	2	3	4	5	6	7	8	9

1	2	2	3	3	3	4	5	7	7
---	---	---	---	---	---	---	---	---	---

sort (HDU 1425)

题目描述:

给你n个整数，请按从大到小的顺序输出其中前m大的数。

输入:

输入有多组数据。每组测试数据有两行，第一行有两个数n,m($0 < n, m < 1000000$), 第二行包含n个各不相同，且都处于区间 $[-500000, 500000]$ 的整数。

输出:

对每组测试数据按从大到小的顺序输出前m大的数。

样例输入:

```
5 3
3 -35 92 213 -644
```

样例输出:

```
213 92 3
```


排序算法的特性

线性排序 → 计数排序

逆序数对 → 归并排序

第K大数 → 快速排序

逆序数对

原序列:

2	8	0	3	4
---	---	---	---	---

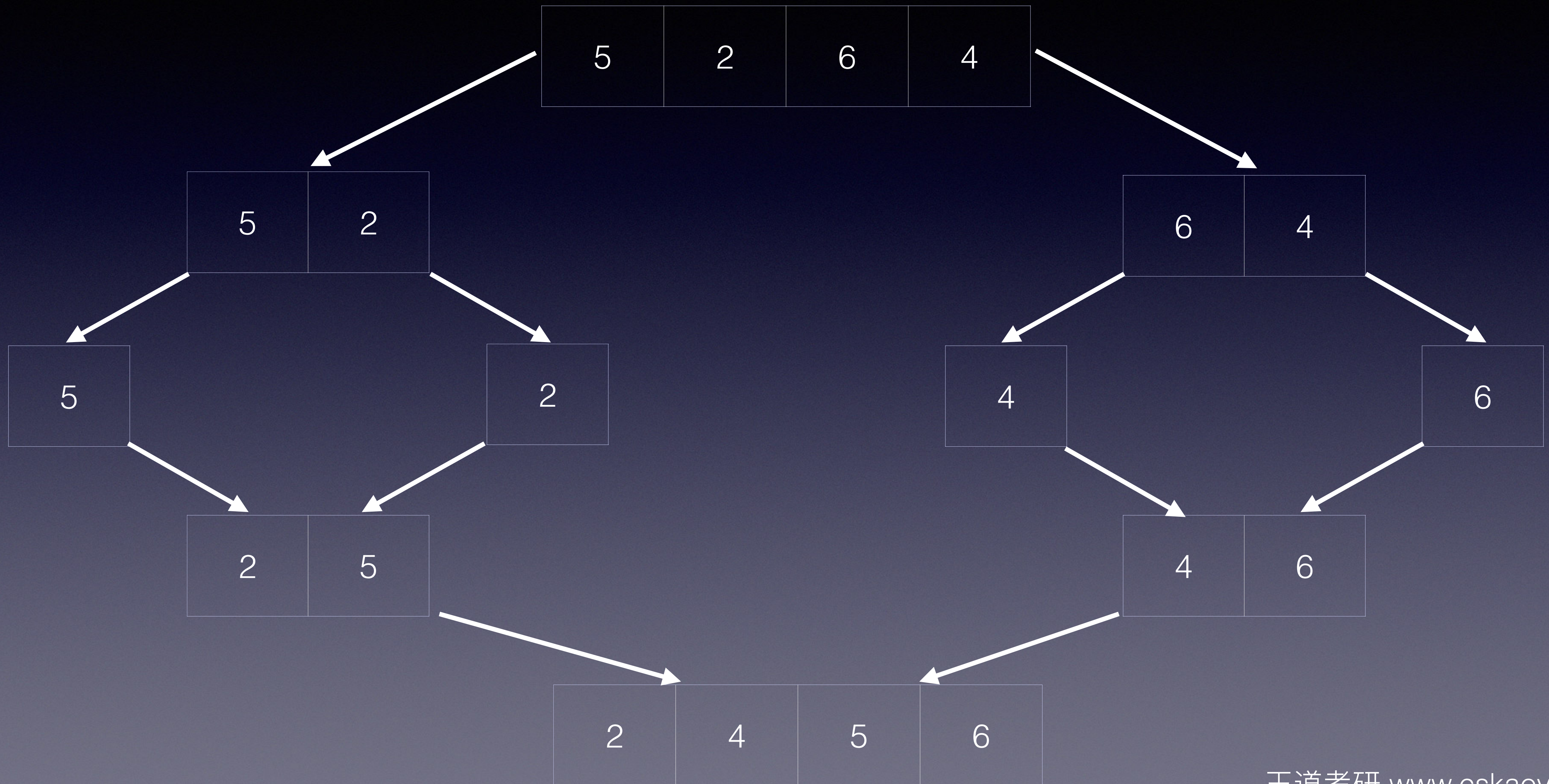
逆序对:

$(2, 0)$ $(8, 0)$ $(8, 3)$ $(8, 4)$

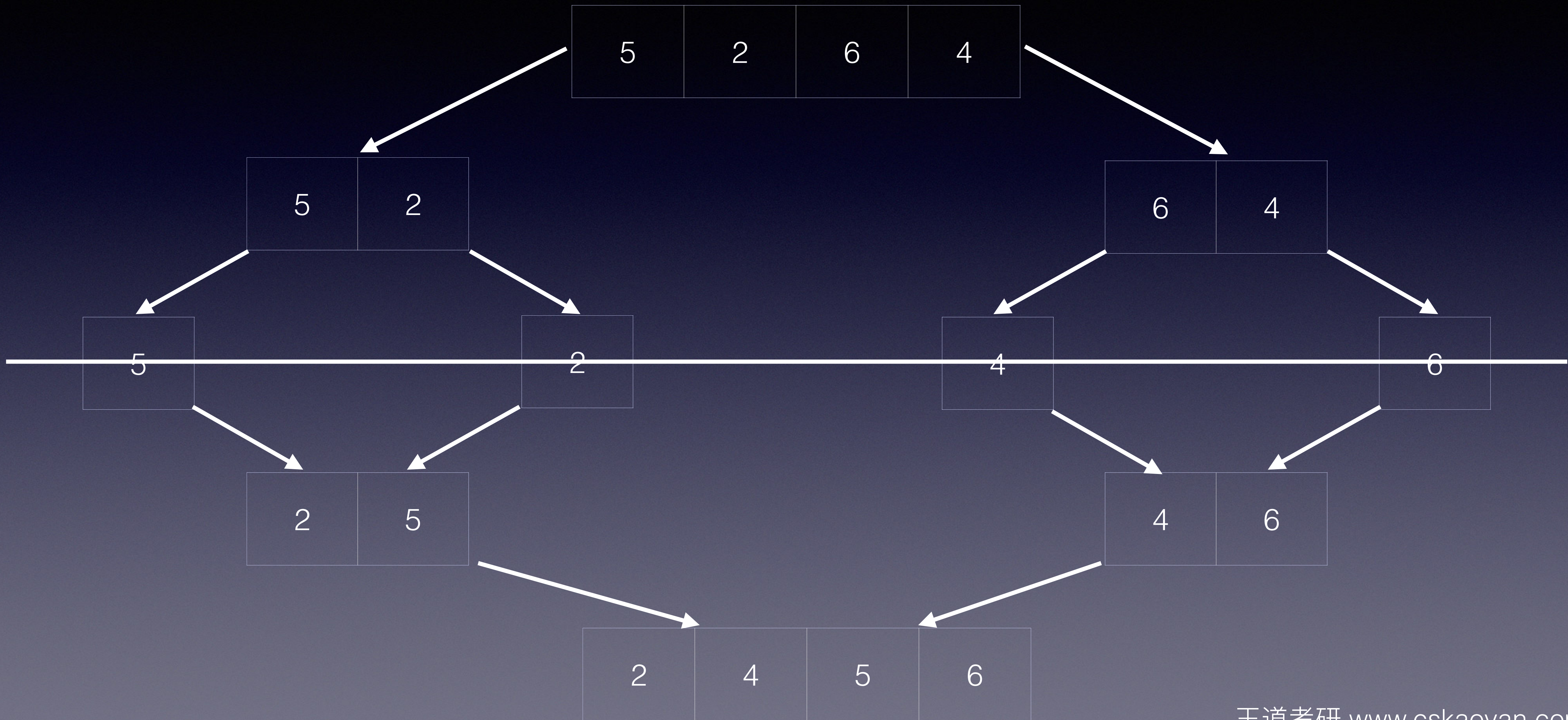
逆序数:

4

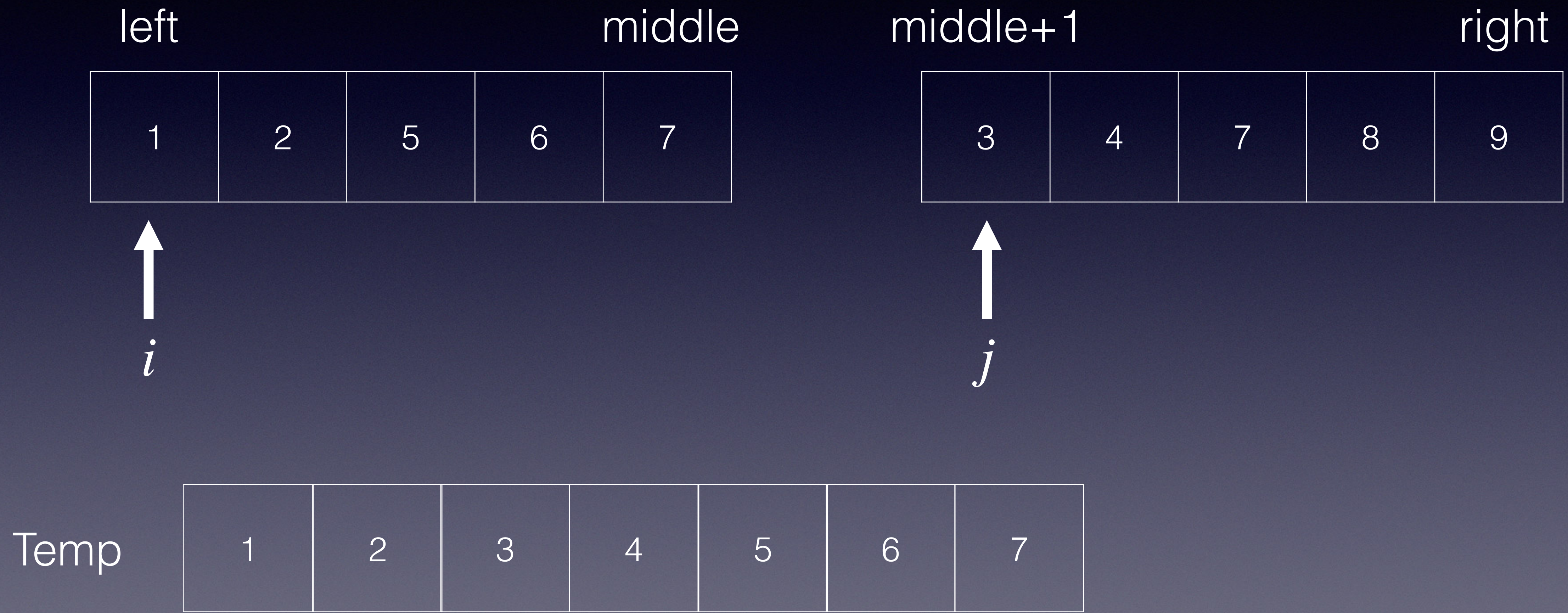
归并排序



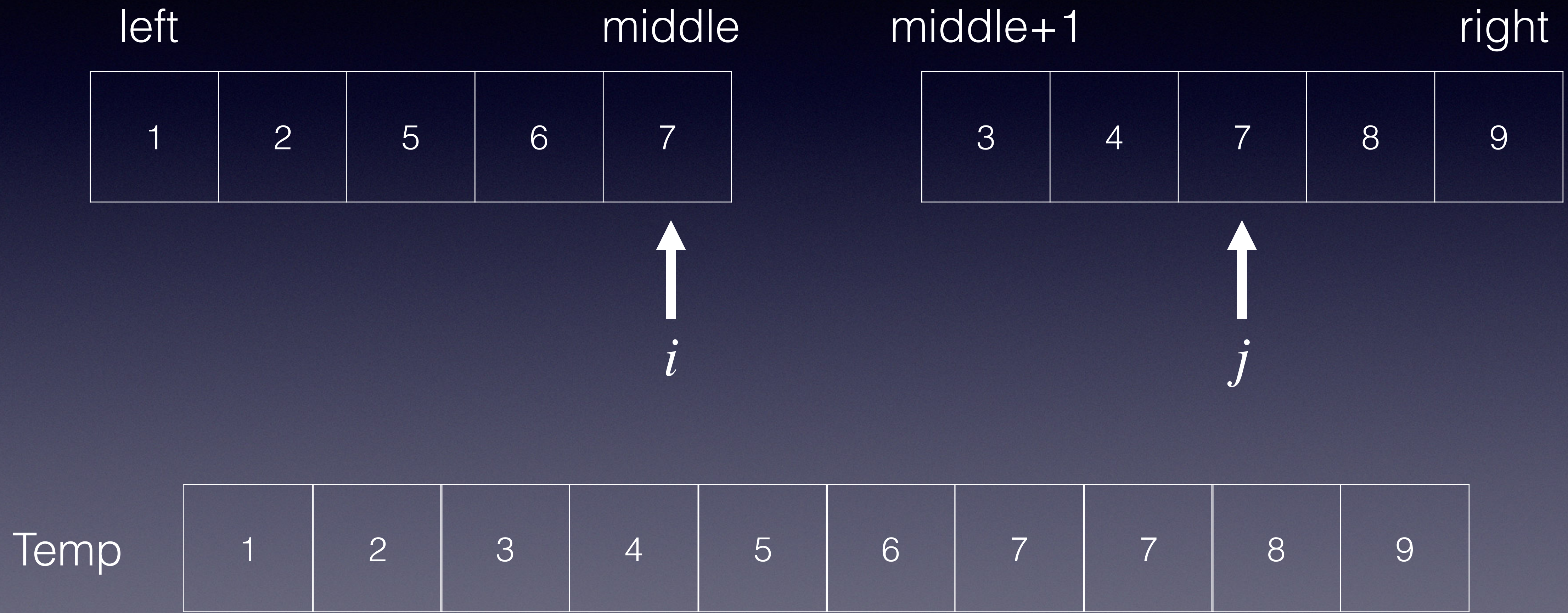
归并排序



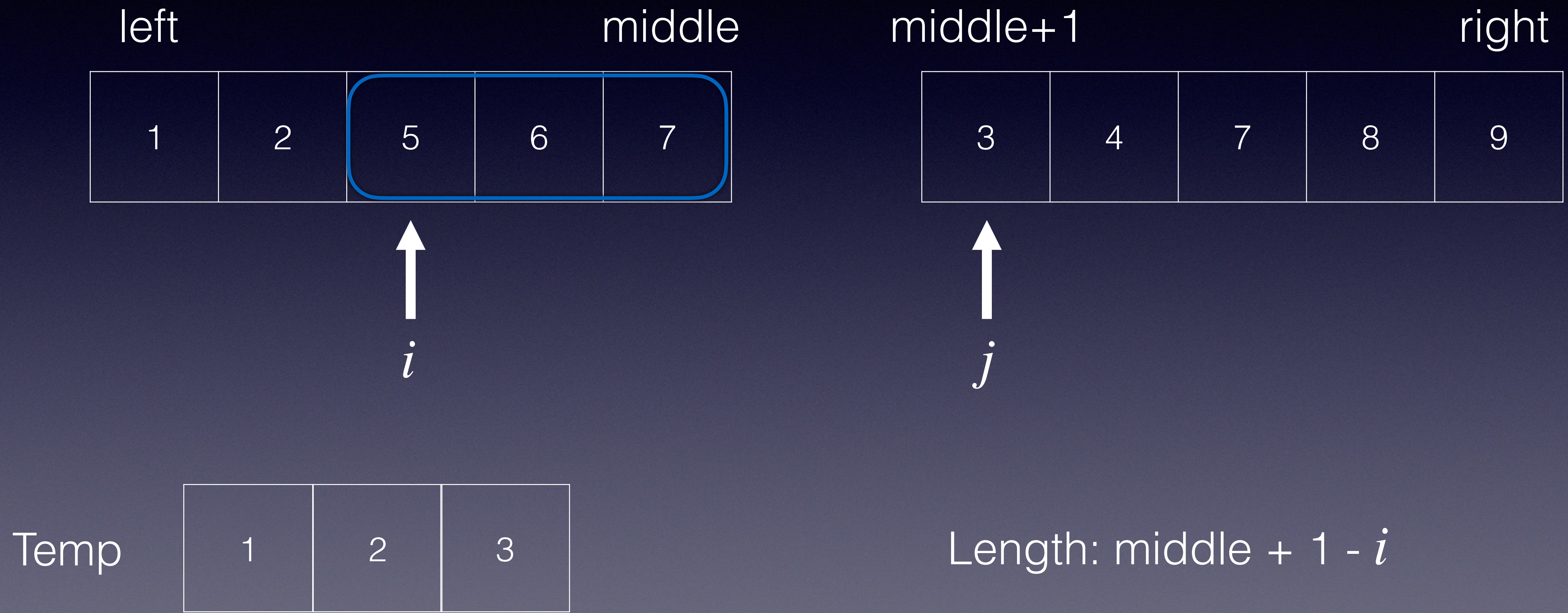
归并排序



归并排序



归并排序



Brainman (POJ 1804)

题目描述:

给你n个整数组成的序列，请求出序列的逆序数。

输入:

输入有多组数据。第一行先给出数据的组数，每组测试数据有第一行，每组数据先给出序列的长度N($0 < N < 100$)，之后输入n个处于区间 $[-1000000, 1000000]$ 的整数。

输出:

对每组测试数据先输出“Scenario #i: ”，在输出该组数据的逆序数，每组输出之间用空行分割开来。

Brainman (POJ 1804)

样例输入：

```
4
4 2 8 0 3
10 0 1 2 3 4 5 6 7 8 9
6 -42 23 6 28 -100 65537
5 0 0 0 0 0
```

样例输出：

```
Scenario #1:
3
```

```
Scenario #2:
0
```

```
Scenario #3:
5
```

```
Scenario #4:
0
```


排序算法的特性

线性排序 → 计数排序

逆序数对 → 归并排序

第K大数 → 快速排序

第K大数

原序列:

8	6	3	4	1	7	2	5
---	---	---	---	---	---	---	---

给定K: 2

K大数: 7

第K大数

原序列:

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

给定K:

2

$n - k$

K大数:

7

$O(n \log n)$

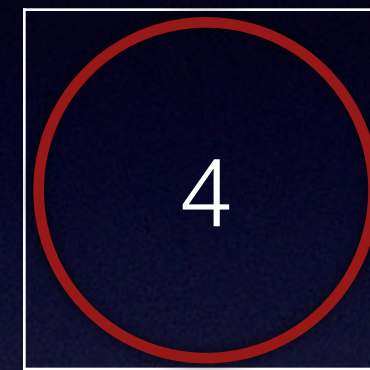
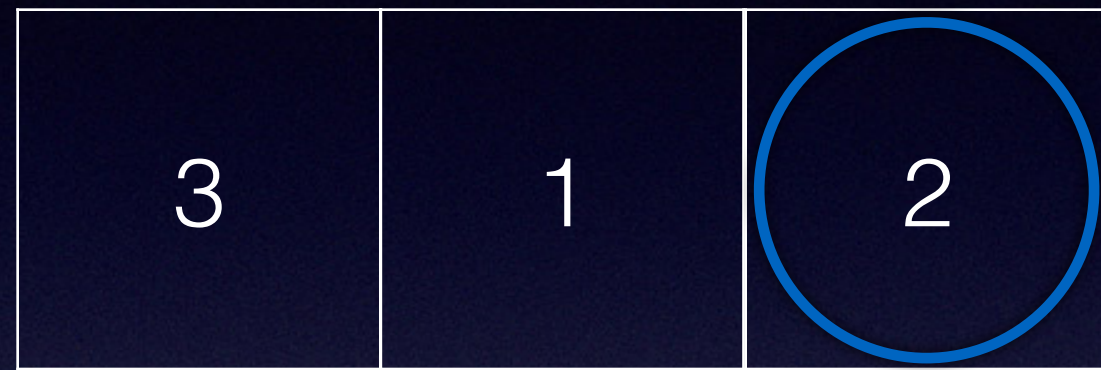
快速排序

Pivot: 4

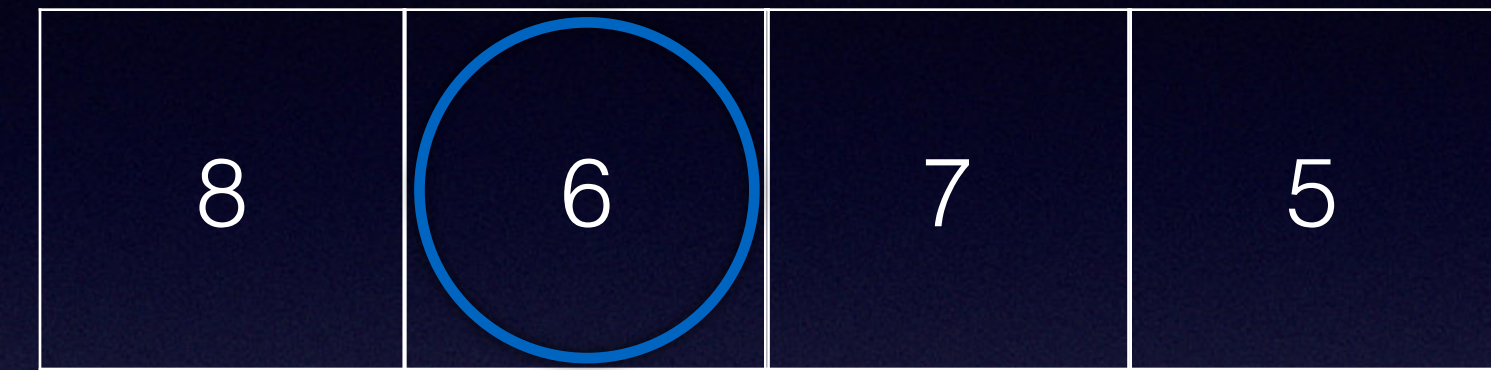
8	6	3	4	1	7	2	5
---	---	---	---	---	---	---	---

快速排序

Pivot: 2



Pivot: 6



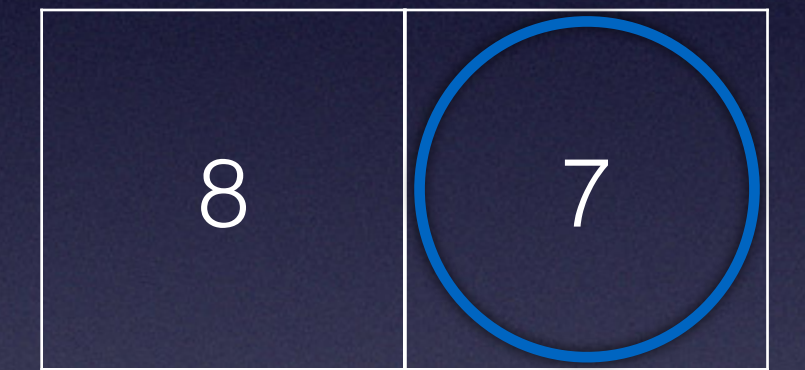
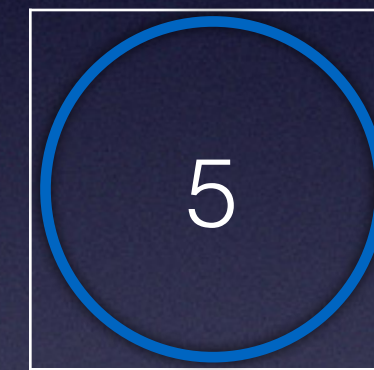
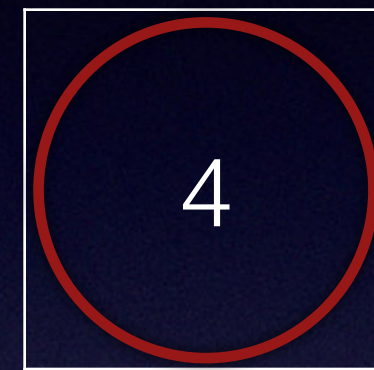
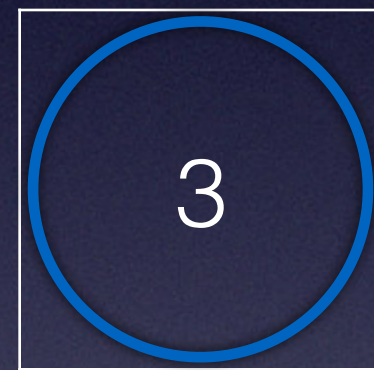
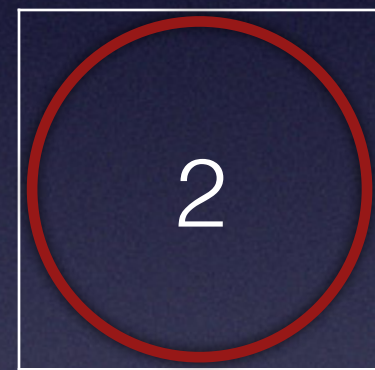
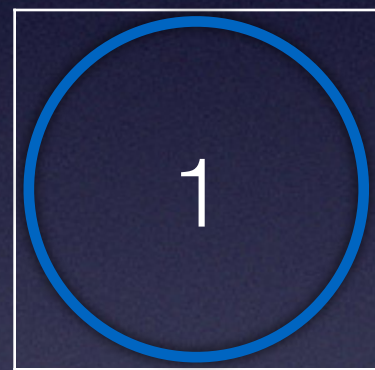
快速排序

Pivot: 1

Pivot: 3

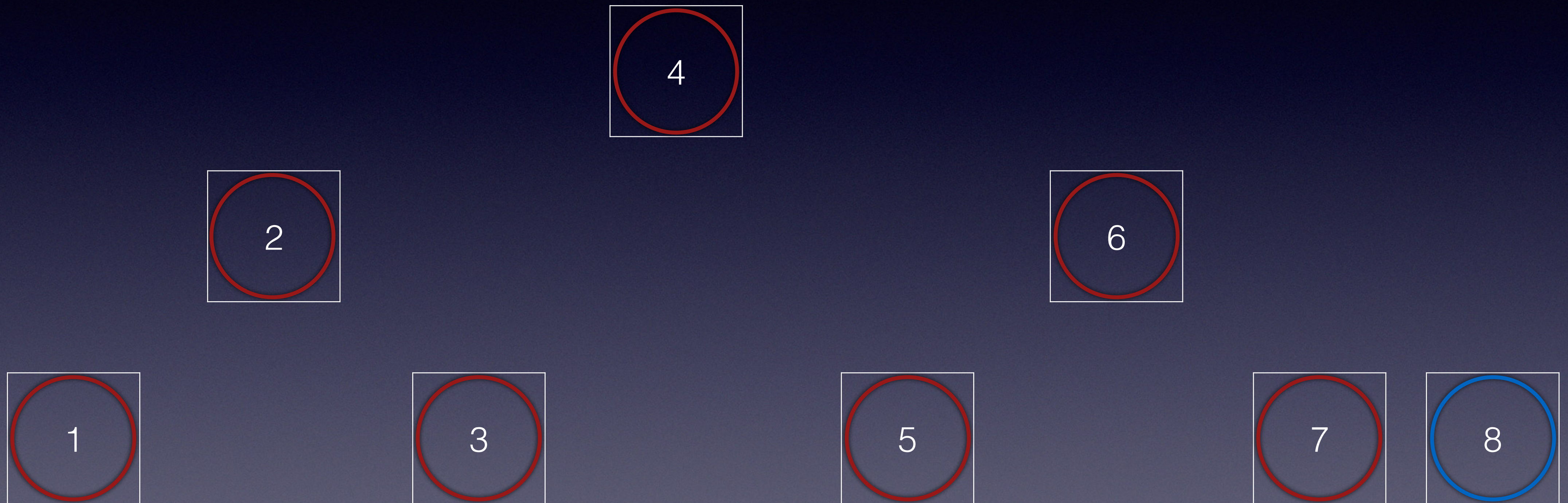
Pivot: 5

Pivot: 7

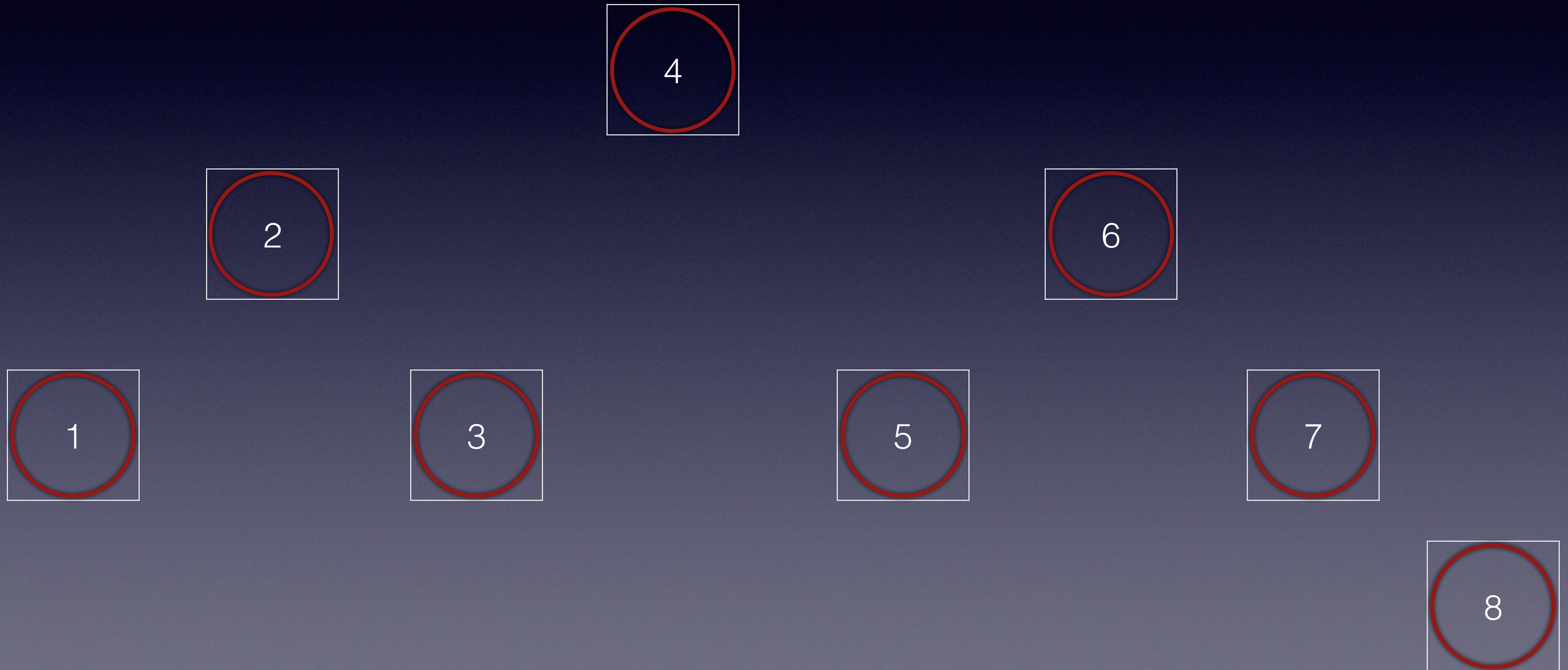


快速排序

Pivot: 8



快速排序



快速排序

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

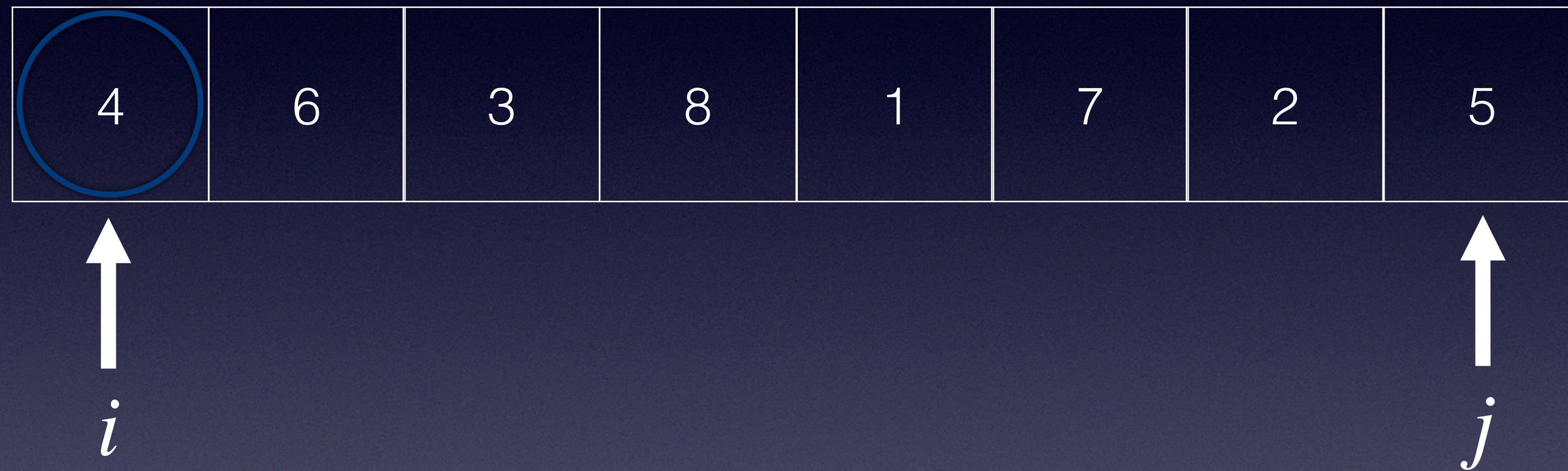
快速排序

Pivot: 4

8	6	3	4	1	7	2	5
---	---	---	---	---	---	---	---

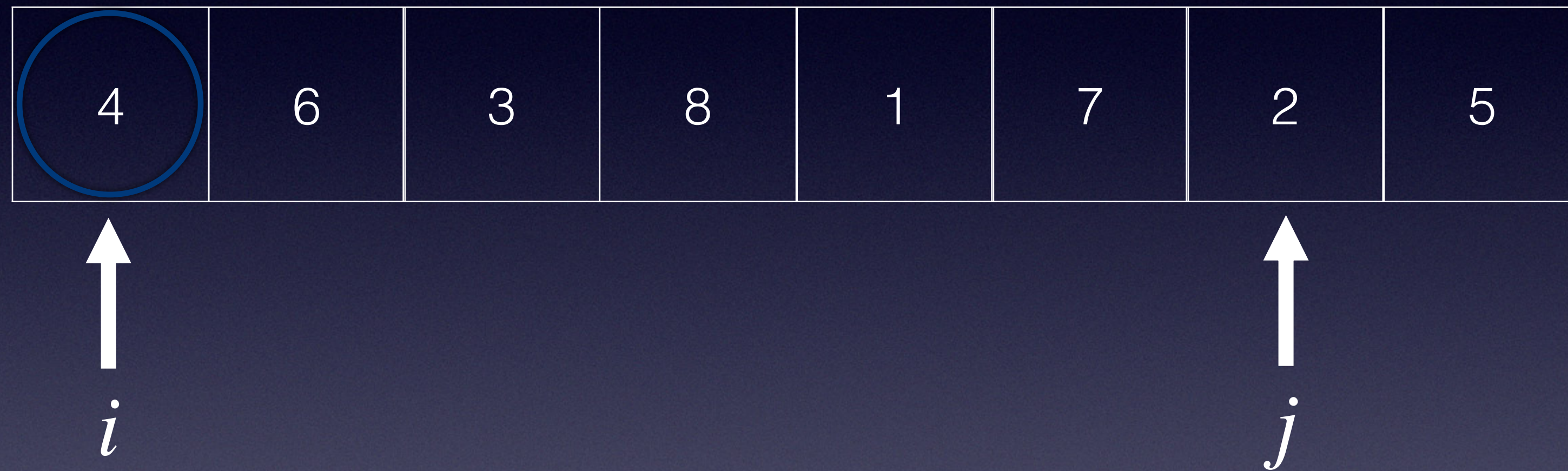
快速排序

Pivot: 4



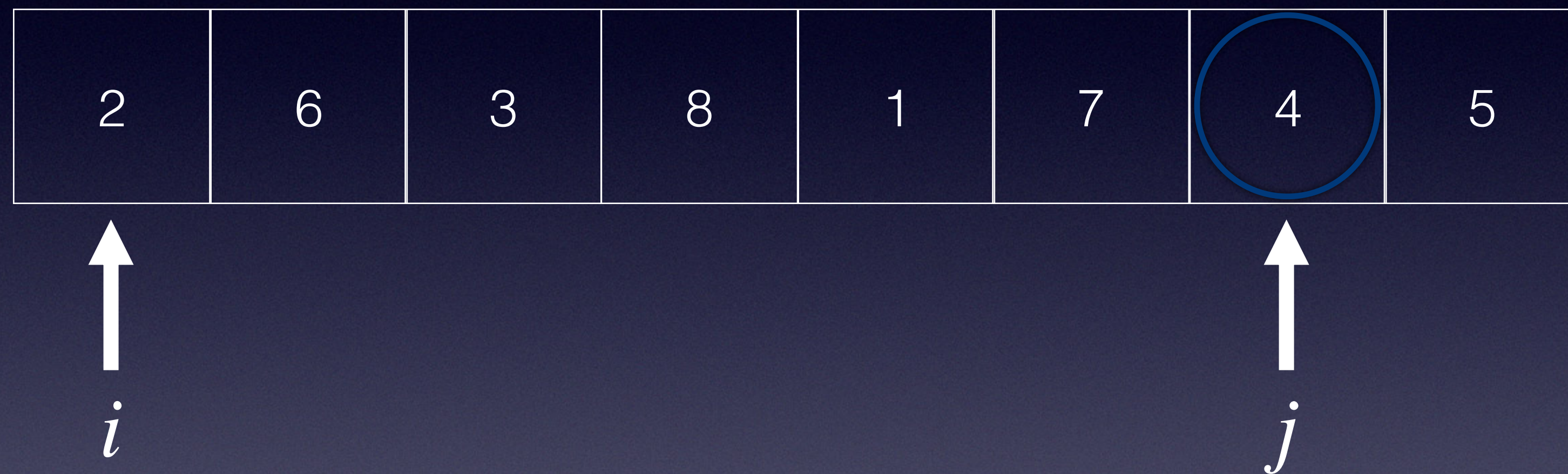
快速排序

Pivot: 4



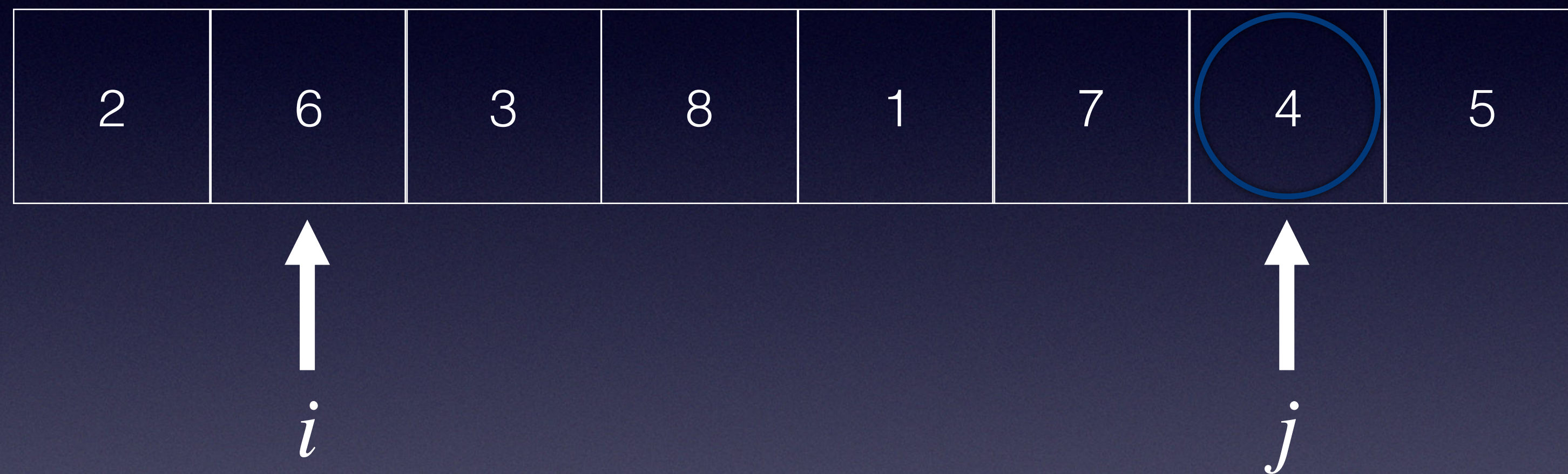
快速排序

Pivot: 4



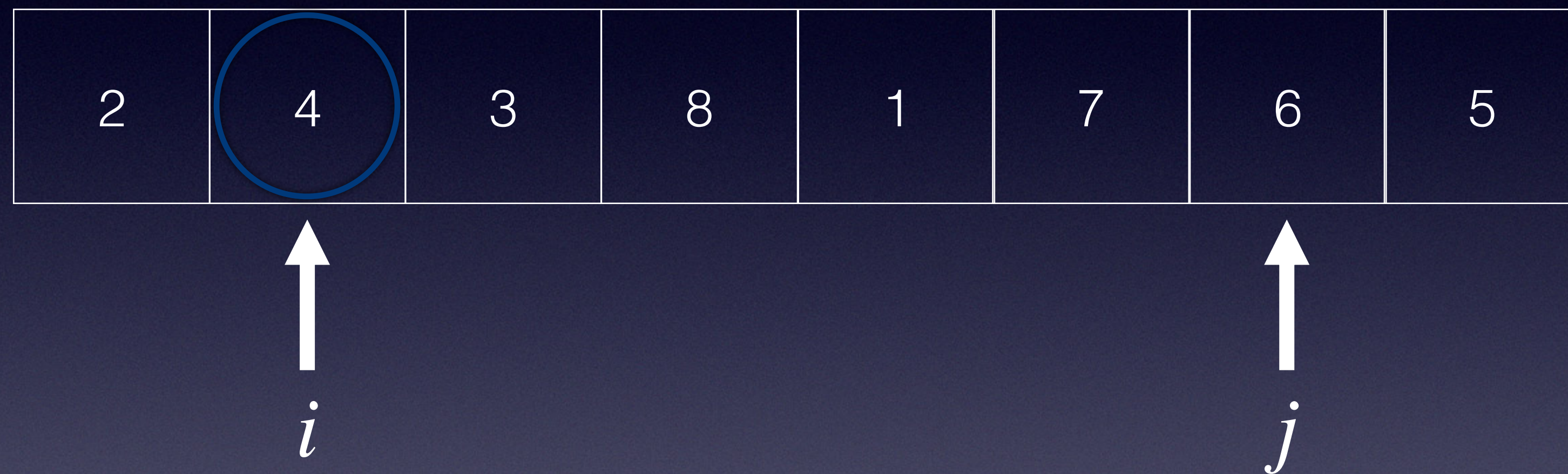
快速排序

Pivot: 4



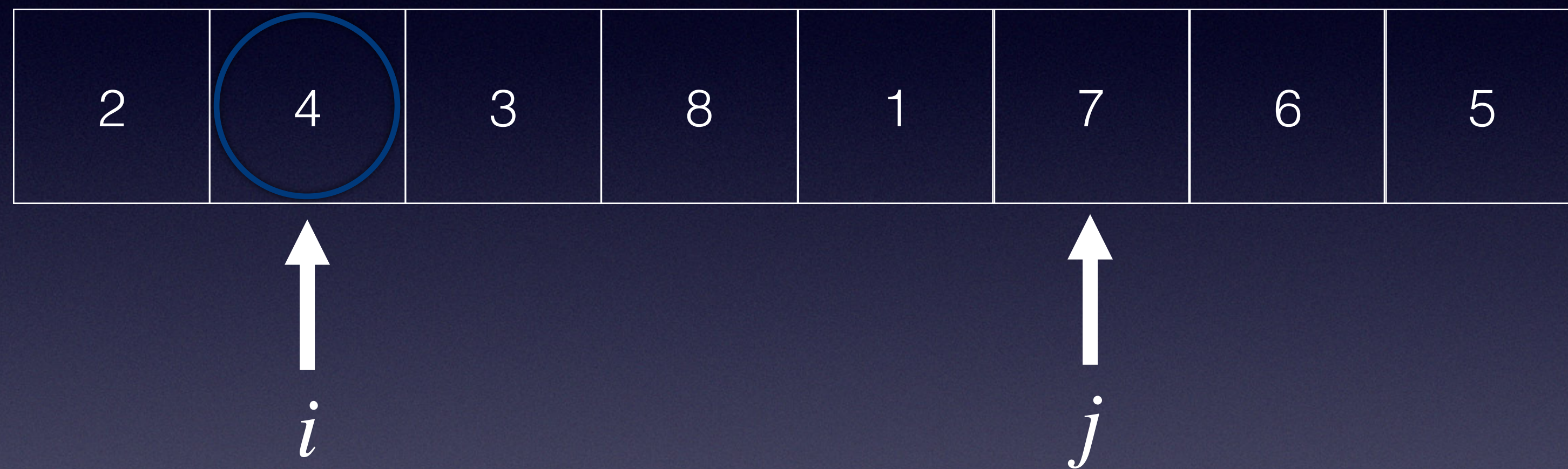
快速排序

Pivot: 4



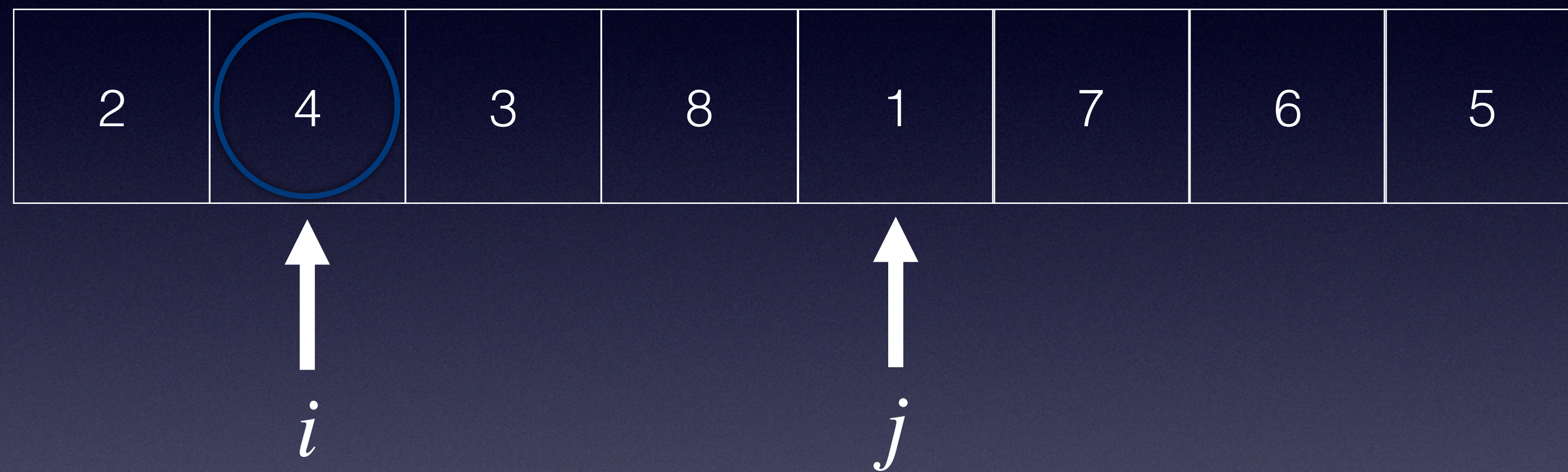
快速排序

Pivot: 4



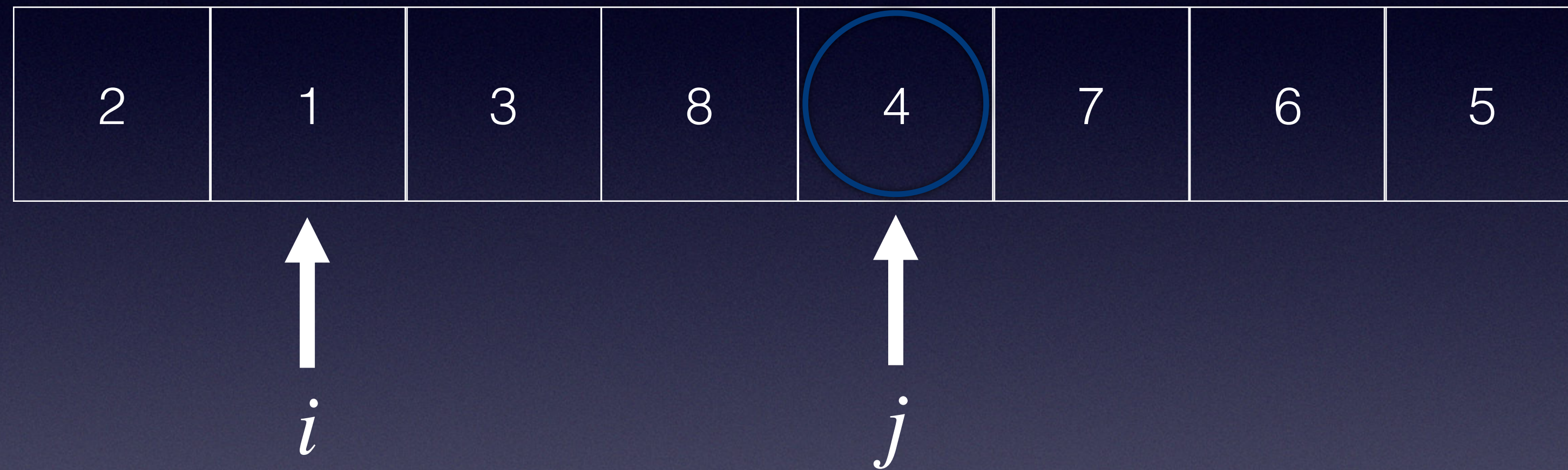
快速排序

Pivot: 4



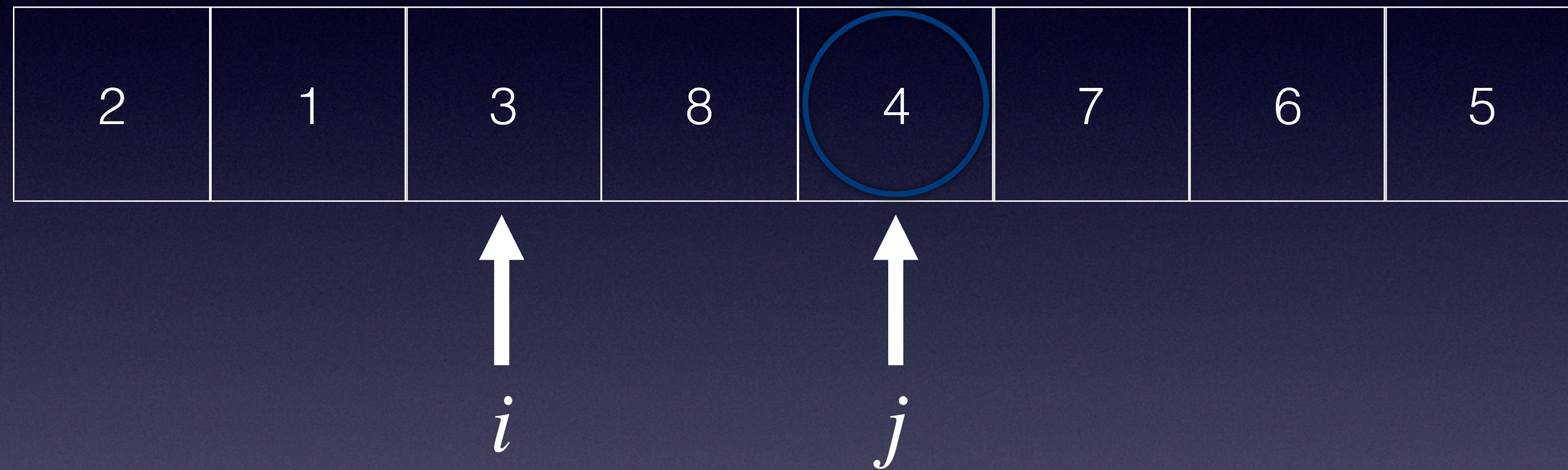
快速排序

Pivot: 4



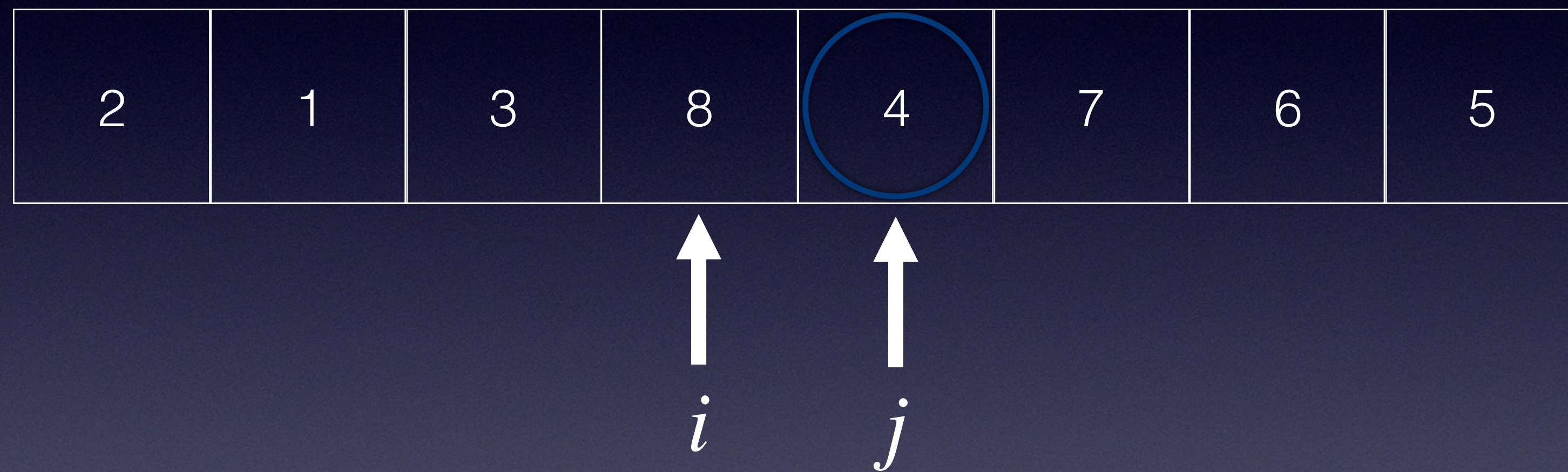
快速排序

Pivot: 4



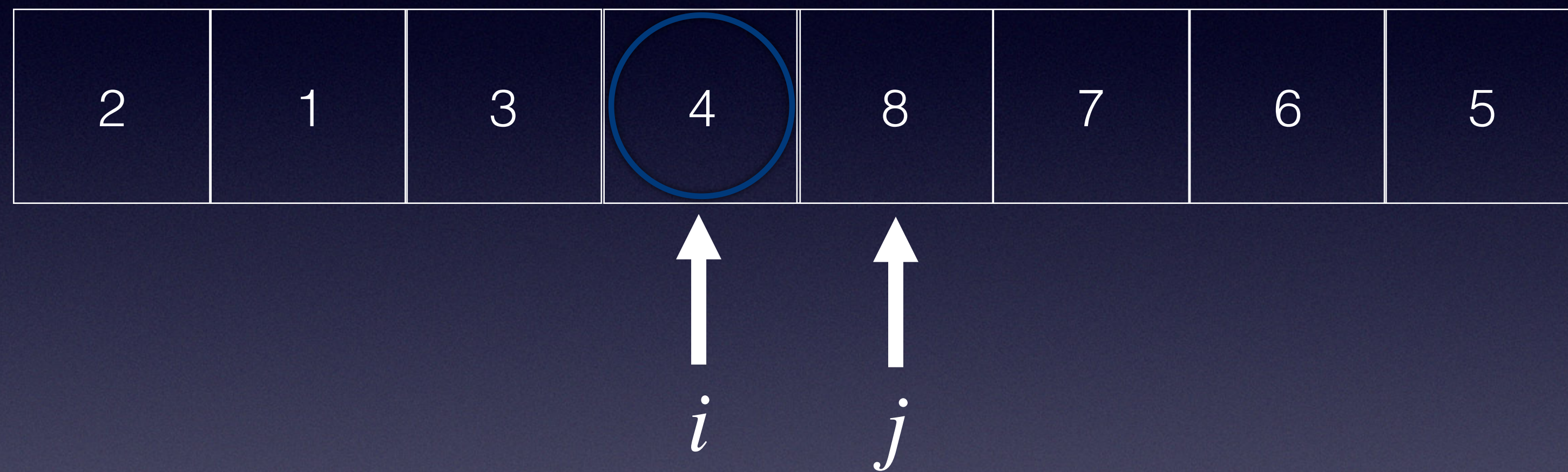
快速排序

Pivot: 4



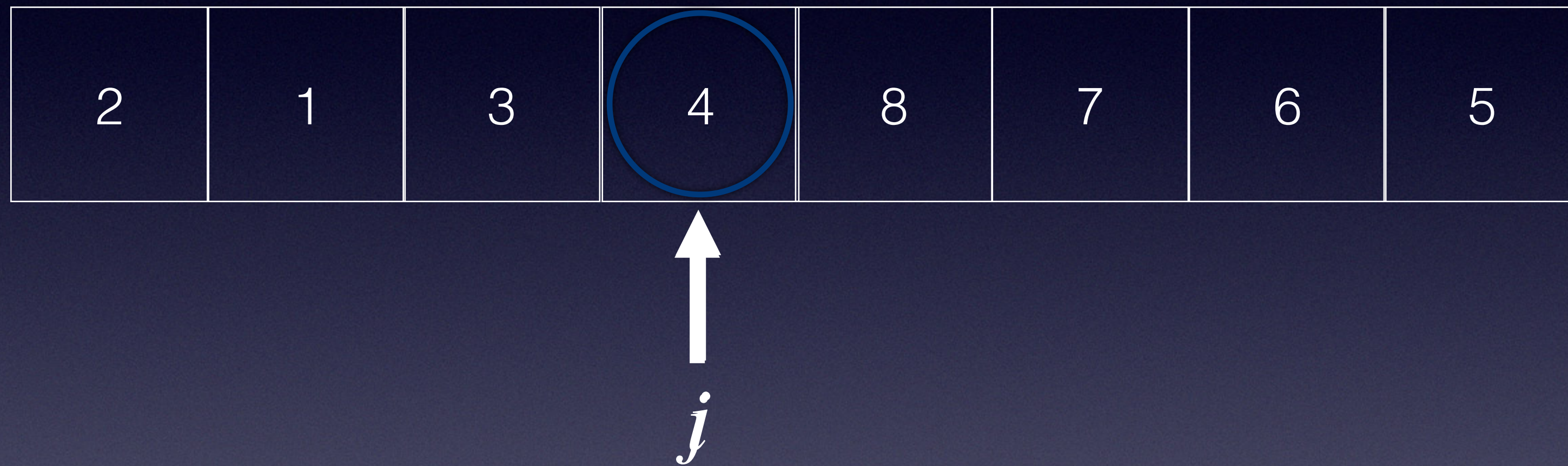
快速排序

Pivot: 4



快速排序

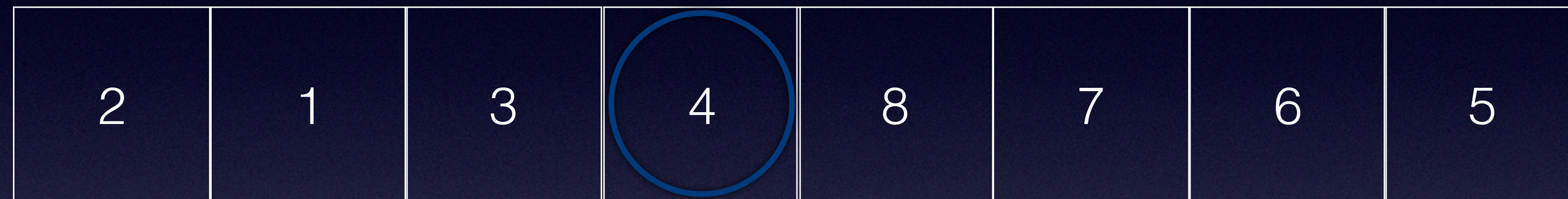
Pivot: 4



快速排序

Pivot: 4

Index: $n - k$



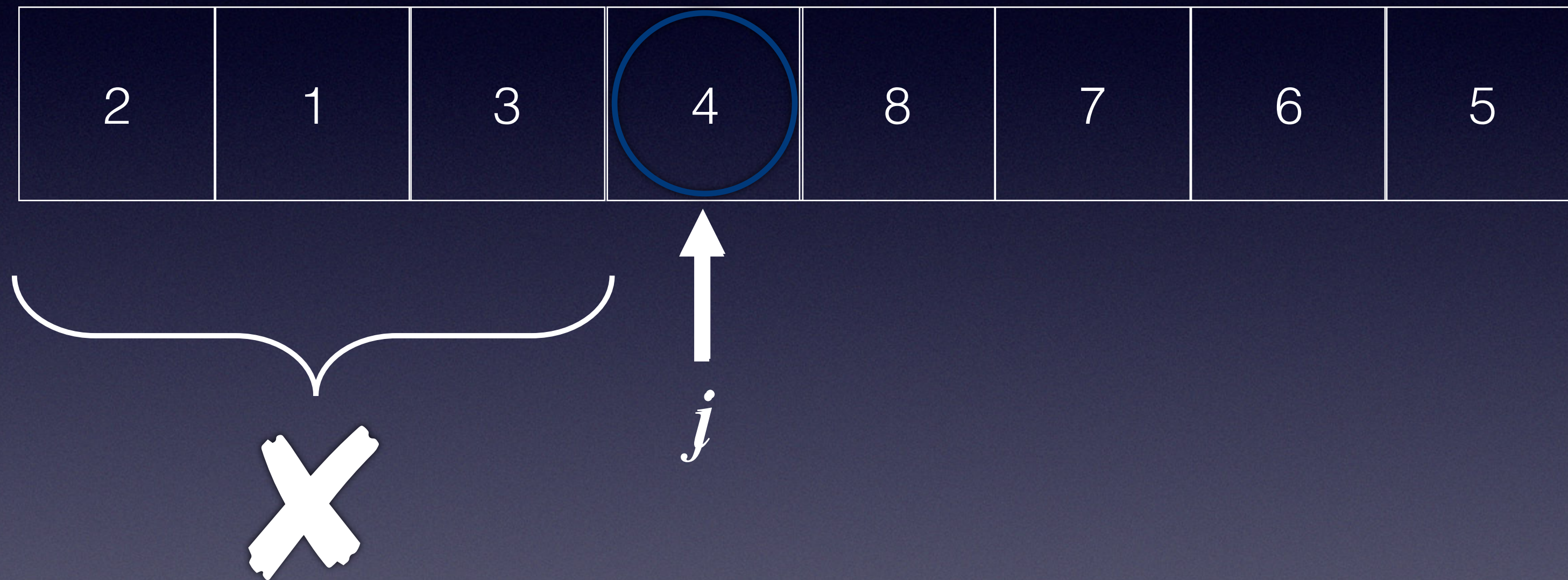
j

$n - k$

快速排序

Pivot: 4

Index: $n - k$

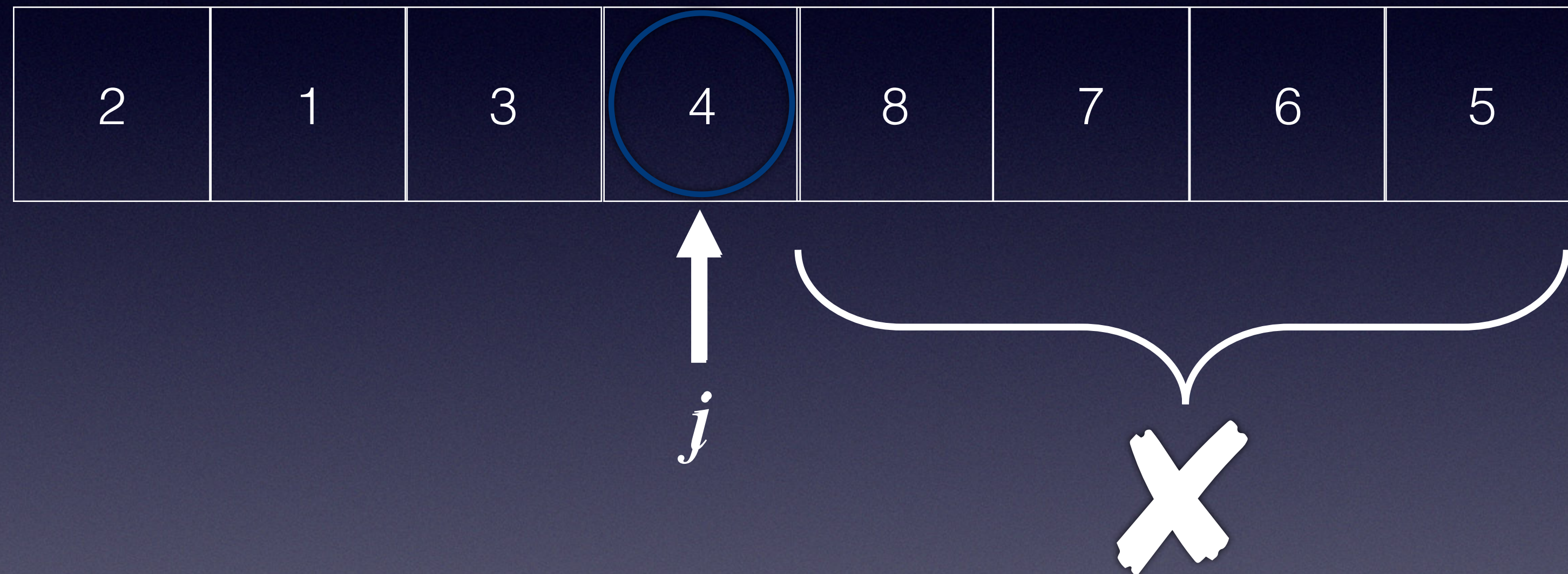


$n - k$

快速排序

Pivot: 4

Index: $n - k$



$n - k$

快速排序

$$n \quad n/2 \quad n/4 \quad \dots \quad 1$$

$$\sum_{i=0}^{\log n} \frac{n}{2^i} = n \sum_{i=0}^{\log n} \frac{1}{2^i} = O(n)$$

Kth Largest Element in an Array (LeetCode 215)

Description:

Find the **k**th largest element in an unsorted array. Note that it is the kth largest element in the sorted order, not the kth distinct element.

Input:

An unsorted array, and a number K ($1 \leq k \leq \text{array's length}$).

Output:

Output the **k**th largest element in an unsorted array.

Sample Input:

[3,2,1,5,6,4] and $k = 2$

[3,2,3,1,2,4,5,5,6] and $k = 4$

Sample Output:

5

4

排序与查找

排序

无序 → 有序

查找

搜索空间中目标值

查找 { 线性查找 $O(n)$
二分查找 $O(\log n)$
散列查找 $O(1)$



线性查找

Target: 5

3	2	2	8	5	7	1	9
---	---	---	---	---	---	---	---



线性查找

Target: 5

3	2	2	8	5	7	1	9
---	---	---	---	---	---	---	---



线性查找

Target: 5

3	2	2	8	5	7	1	9
---	---	---	---	---	---	---	---



线性查找

Target: 5

3	2	2	8	5	7	1	9
---	---	---	---	---	---	---	---



线性查找

Target: 5

3	2	2	8	5	7	1	9
---	---	---	---	---	---	---	---



例题3.5 查找

题目描述：

输入数组长度 n ，输入数组 $a[1...n]$ ，输入查找次数 m ，输入查找数字 $b[1...m]$ ，输出YES或NO，找到则输出YES，否则输出NO。

输入：

输入有多组数据。每组输入 n ，然后输入 n 个整数，再输入 m ，最后输入 m 个整数。

输出：

若在 n 个数组中，则输出YES，否则输出NO。

样例输入：

```
5
1 5 2 4 3
3
2 5 6
```

样例输出：

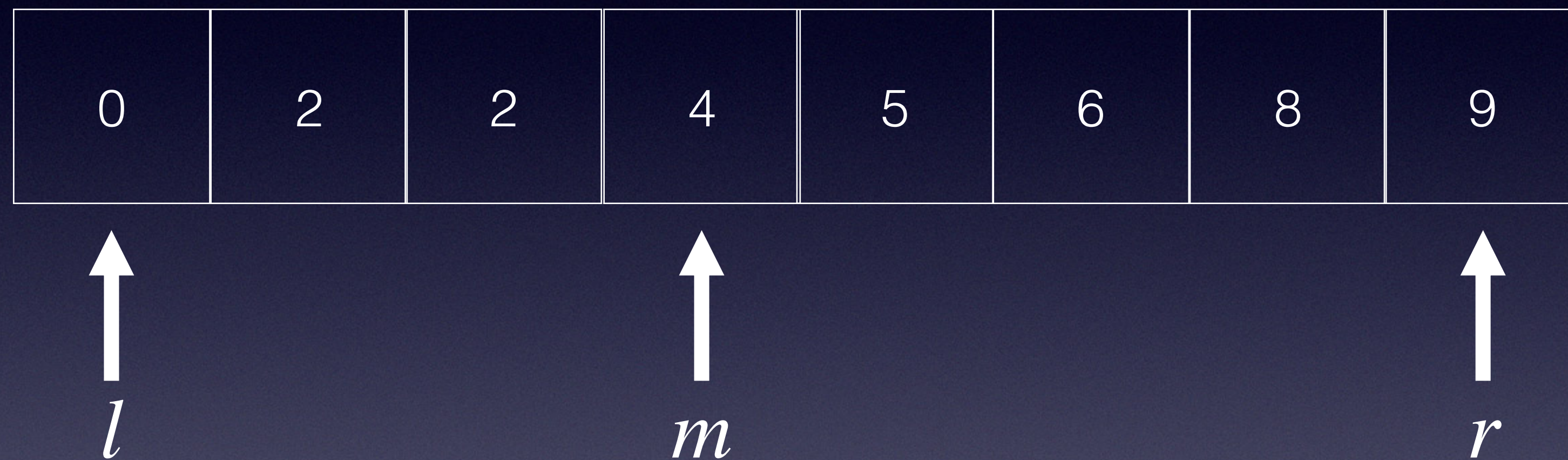
```
YES
YES
NO
```





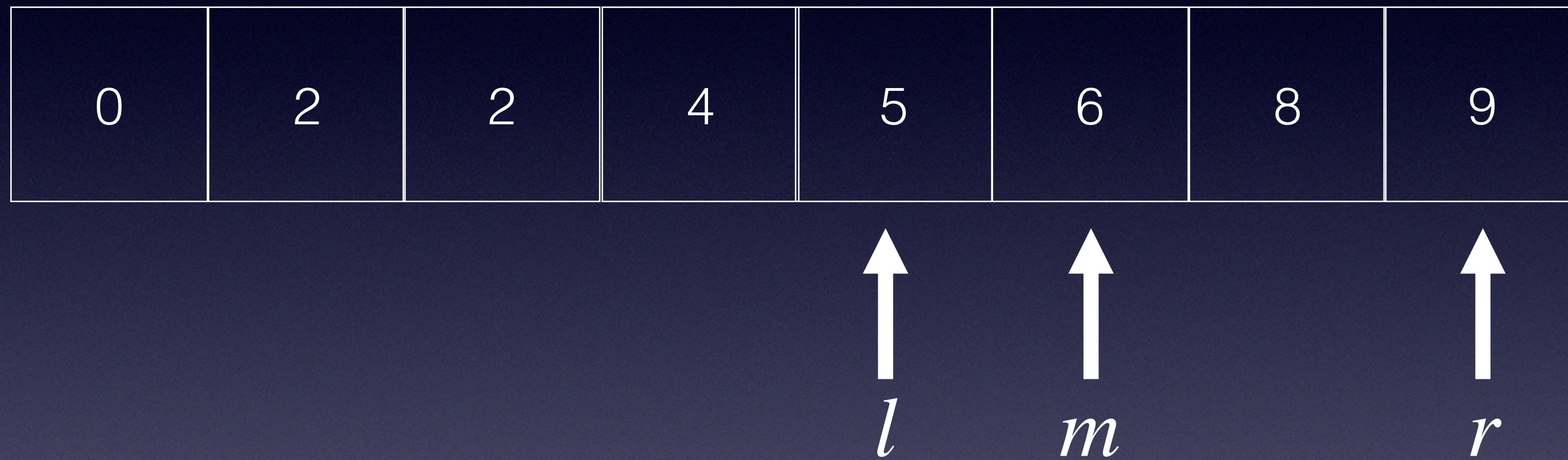

二分查找

Target: 5



二分查找

Target: 5



二分查找

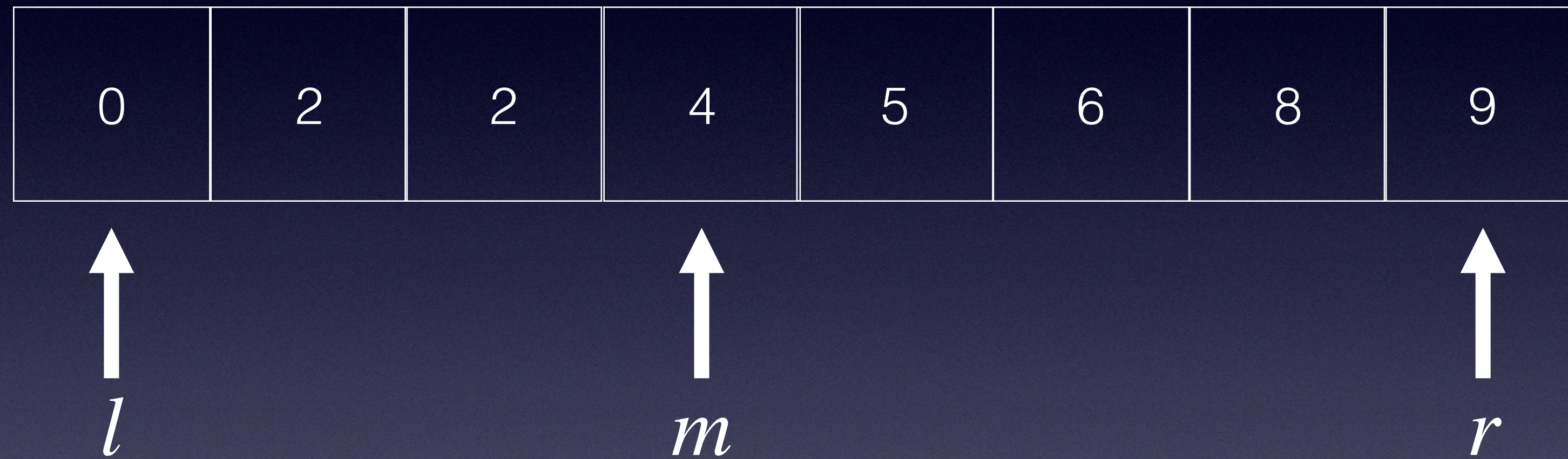
Target: 5

0	2	2	4	5	6	8	9
---	---	---	---	---	---	---	---



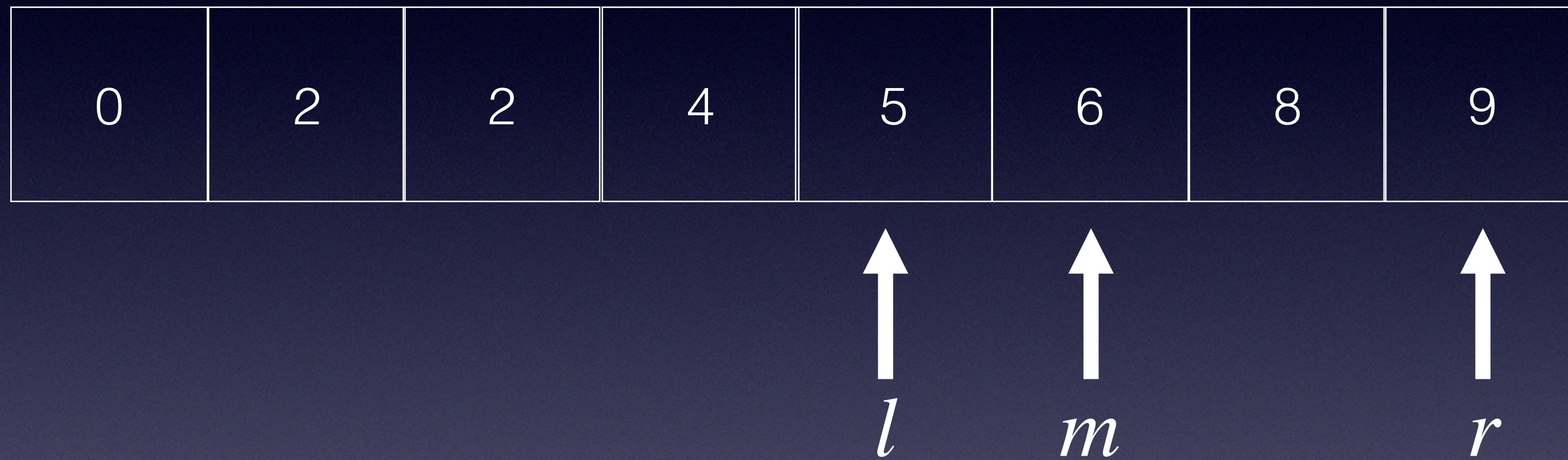
二分查找

Target: 7



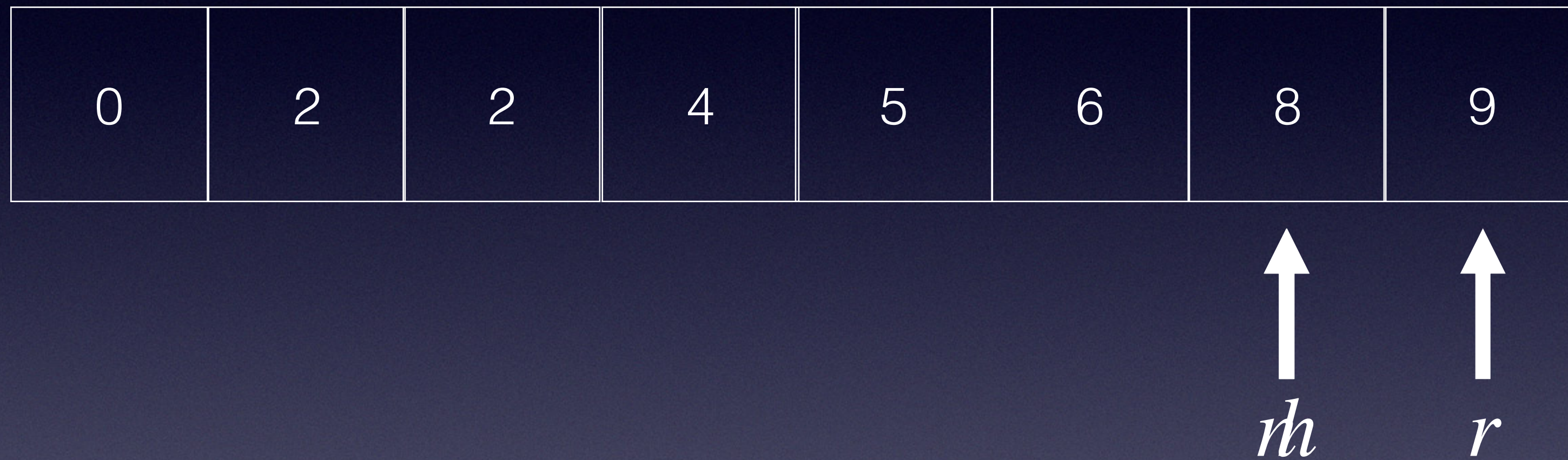
二分查找

Target: 7



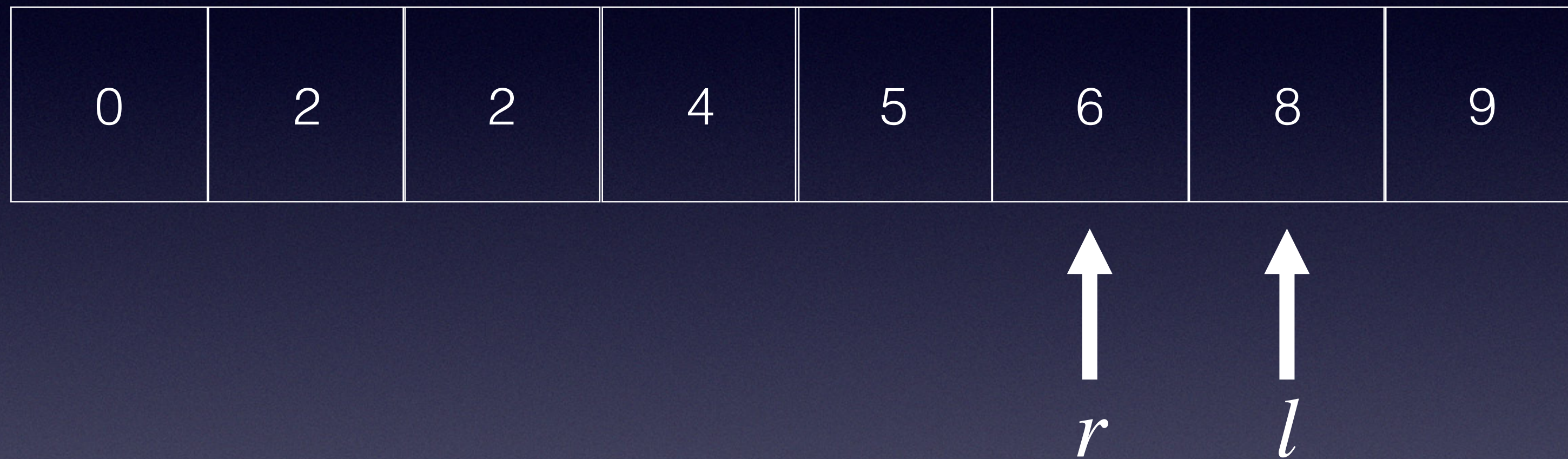
二分查找

Target: 7



二分查找

Target: 7



例题3.5 查找

题目描述：

输入数组长度 n ，输入数组 $a[1...n]$ ，输入查找次数 m ，输入查找数字 $b[1...m]$ ，输出YES或NO，找到则输出YES，否则输出NO。

输入：

输入有多组数据。每组输入 n ，然后输入 n 个整数，再输入 m ，最后输入 m 个整数。

输出：

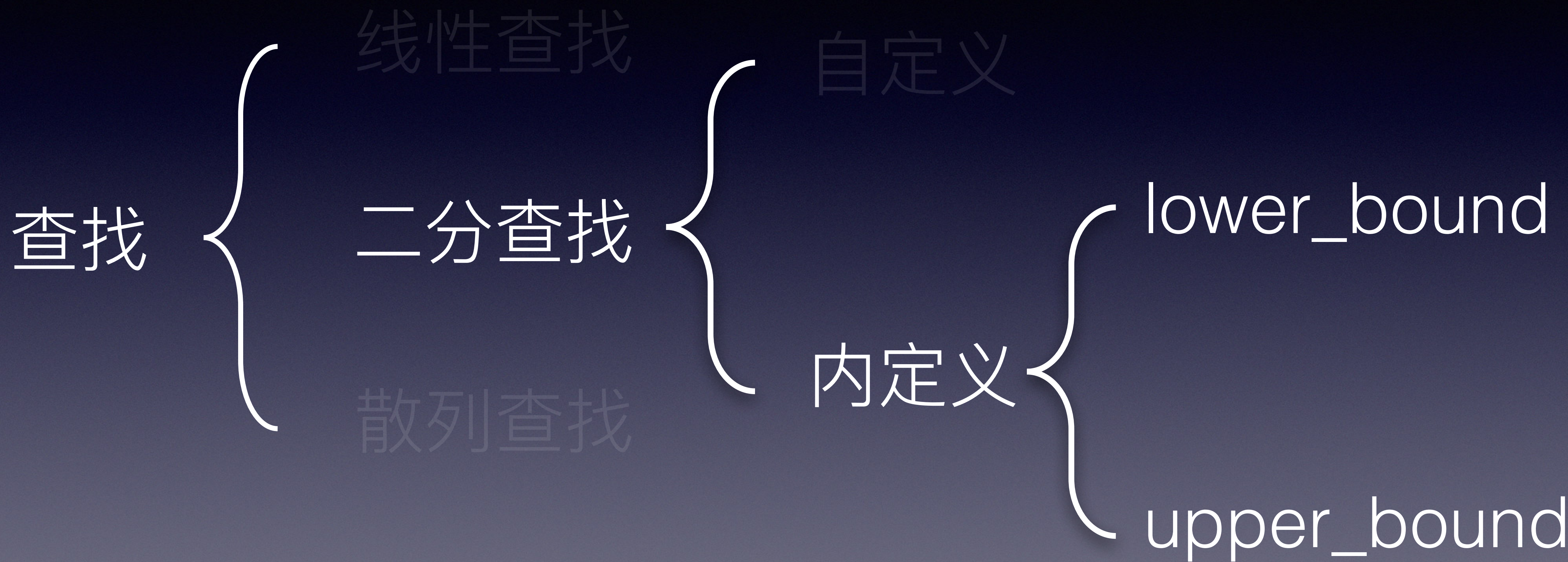
若在 n 个数组中，则输出YES，否则输出NO。

样例输入：

```
5
1 5 2 4 3
3
2 5 6
```

样例输出：

```
YES
YES
NO
```

lower_bound

返回大于或等于目标值的第一个位置

upper_bound

返回大于目标值的第一个位置

Target:4

lower_bound



1	2	4	4	5	8	10	22
---	---	---	---	---	---	----	----



upper_bound

Target:7

lower_bound



1	2	4	4	5	8	10	22
---	---	---	---	---	---	----	----



upper_bound

Target:30

lower_bound



1	2	4	4	5	8	10	22
---	---	---	---	---	---	----	----



upper_bound

例题3.5 查找

题目描述：

输入数组长度 n ，输入数组 $a[1...n]$ ，输入查找次数 m ，输入查找数字 $b[1...m]$ ，输出YES或NO，找到则输出YES，否则输出NO。

输入：

输入有多组数据。每组输入 n ，然后输入 n 个整数，再输入 m ，最后输入 m 个整数。

输出：

若在 n 个数组中，则输出YES，否则输出NO。

样例输入：

```
5
1 5 2 4 3
3
2 5 6
```

样例输出：

```
YES
YES
NO
```






散列查找

Value

Index

2	6	4	8	3	5
0	1	2	3	4	5

$$H(Value) = Index$$

$$H(2) = 0 \quad H(6) = 1 \dots$$

散列查找

Value

Index

2	6	4	8	3	5
0	1	2	3	4	5

Value

Index

		2	3	4	5	6		8
0	1	2	3	4	5	6	7	8

$$H(Value) = Value$$

散列查找

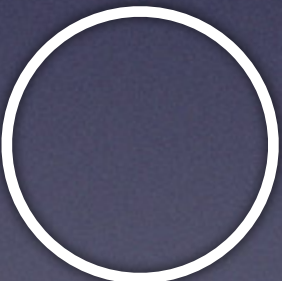
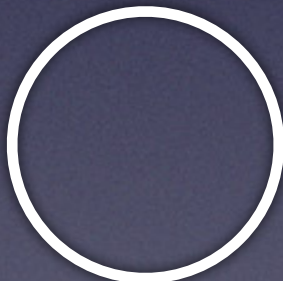
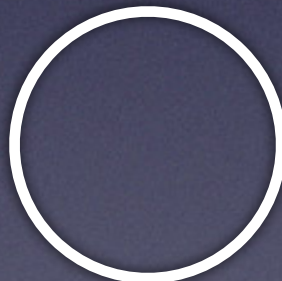
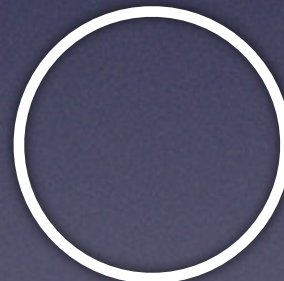
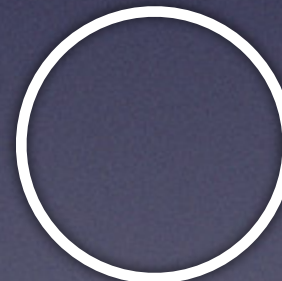

Value

Index

2	6	4	8	3	5
0	1	2	3	4	5

Value

Index

								
0	1	2	3	4	5	6	7	8

$$H(Value) = True$$

例题3.5 查找

题目描述：

输入数组长度 n ，输入数组 $a[1...n]$ ，输入查找次数 m ，输入查找数字 $b[1...m]$ ，输出YES或NO，找到则输出YES，否则输出NO。

输入：

输入有多组数据。每组输入 n ，然后输入 n 个整数，再输入 m ，最后输入 m 个整数。

输出：

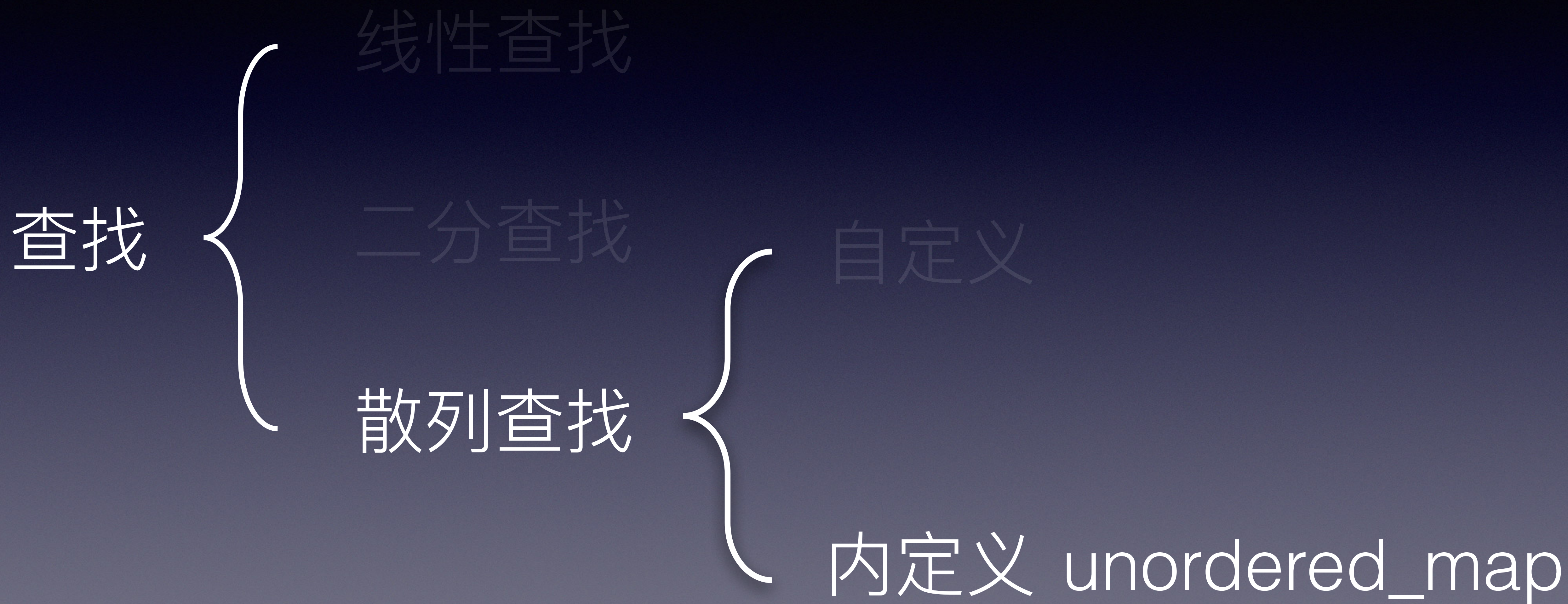
若在 n 个数组中，则输出YES，否则输出NO。

样例输入：

```
5
1 5 2 4 3
3
2 5 6
```

样例输出：

```
YES
YES
NO
```

例题3.5 查找

题目描述：

输入数组长度 n ，输入数组 $a[1...n]$ ，输入查找次数 m ，输入查找数字 $b[1...m]$ ，输出YES或NO，找到则输出YES，否则输出NO。

输入：

输入有多组数据。每组输入 n ，然后输入 n 个整数，再输入 m ，最后输入 m 个整数。

输出：

若在 n 个数组中，则输出YES，否则输出NO。

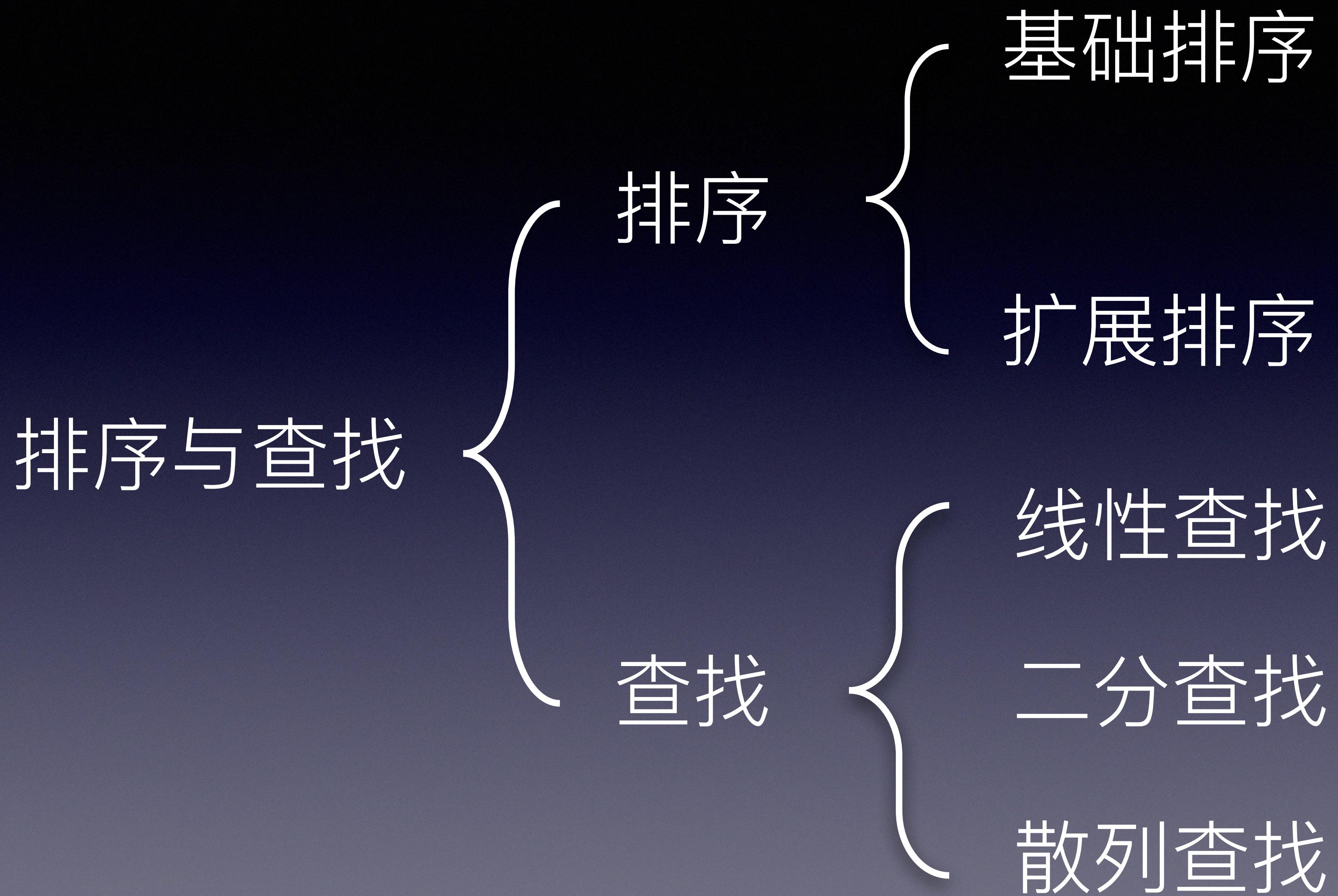
样例输入：

```
5
1 5 2 4 3
3
2 5 6
```

样例输出：

```
YES
YES
NO
```


查找 { 线性查找
二分查找
散列查找



王道考研-机试指南

第三章 排序与查找

王道考研-机试指南

www.cskaoyan.com