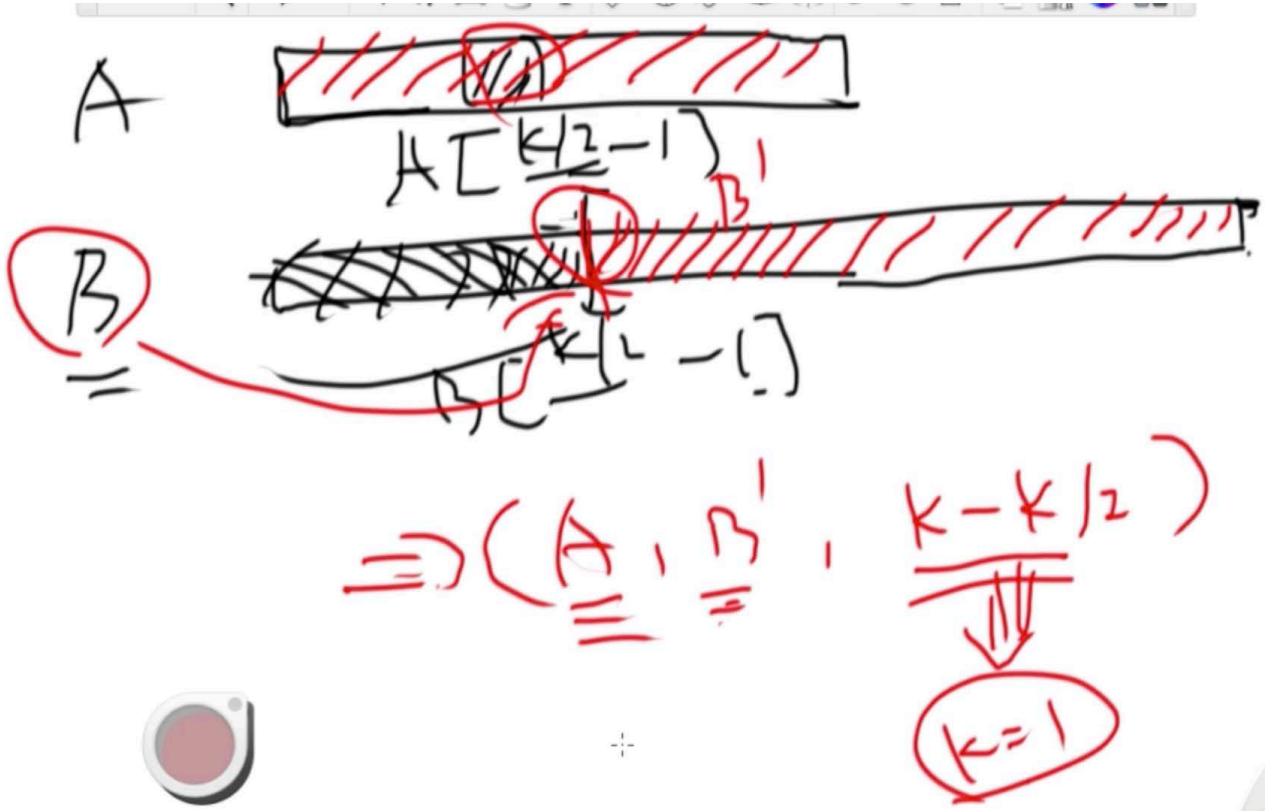
Median of Two Sorted Arrays

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

http://www.jiuzhang.com/solutions/median-of-two-sorted-arrays/



A

B

11 13 14 18

A [2]

B [2]

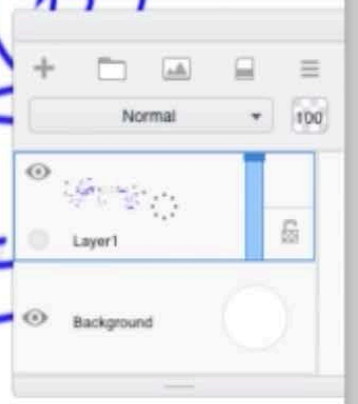
$\frac{3}{2} = 1$

$k = 2$

$\frac{k}{2} = 1$

$k = 6$

$k =$



461. Kth Smallest Numbers in Unsorted Array

Description | Notes | Testcase | Judge

Find the kth smallest numbers in an unsorted integer array.

Have you met this question in a real interview? Yes

Example

Given `[3, 4, 1, 2, 5]`, k = `3`, the 3rd smallest numbers are `[1, 2, 3]`.

Challenge

Tags

Related Problems

```java
class Solution {
    /*
     * @param k an integer
     * @param nums an integer array
     * @return kth smallest element
     */
    public int kthSmallest(int k, int[] nums) {
        // write your code here
        return quickSelect(nums, 0, nums.length - 1, k - 1);
    }

    public int quickSelect(int[] A, int start, int end , int k) {

        if (start == end)
            return A[start];

        int left = start, right = end;
        int pivot = A[(start + end) / 2];

        while (left <= right) {
            while (left <= right && A[left] < pivot) {
                left++;
            }

            while (left <= right && A[right] > pivot) {
                right--;
            }
            if (left <= right) {
                int temp = A[left];
                A[left] = A[right];
                A[right] = temp;

                left++;
                right--;
            }
        }

        if (right >= k && start <= right)
            return quickSelect(A, start, right, k);
        else if (left <= k && left <= end)
            return quickSelect(A, left, end, k);
```

80. Median

Description | Notes | Testcase | Judge

Given a unsorted array with integers, find the median of it.

A median is the middle number of the array after it is sorted.

If there are even numbers in the array, return the `N/2` -th number after sorted.

Have you met this question in a real interview? Yes

Example

Given `[4, 5, 1, 2, 3]`, return `3`.

Given `[7, 9, 4, 5]`, return `5`.

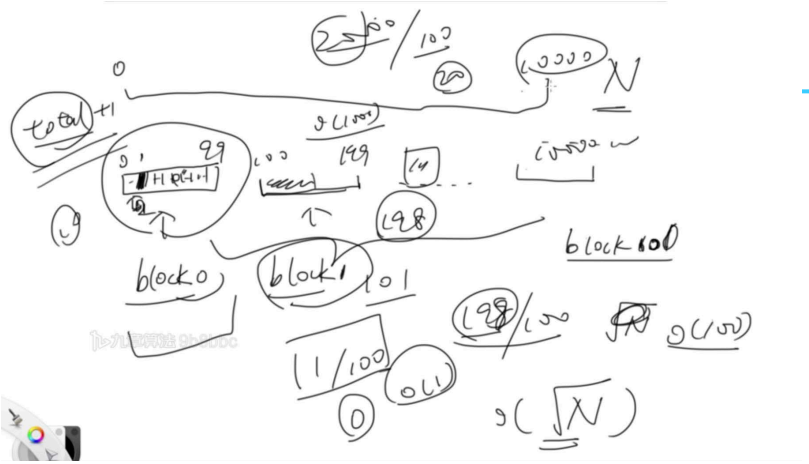
Challenge

Tags

Related Problems

```java
public class Solution {
    /**
     * @param nums: A list of integers.
     * @return: An integer denotes the middle number of the array.
     */
    public int median(int[] nums) {
        // write your code here

        findKth(nums, nums.length / 2);
    }
}
```

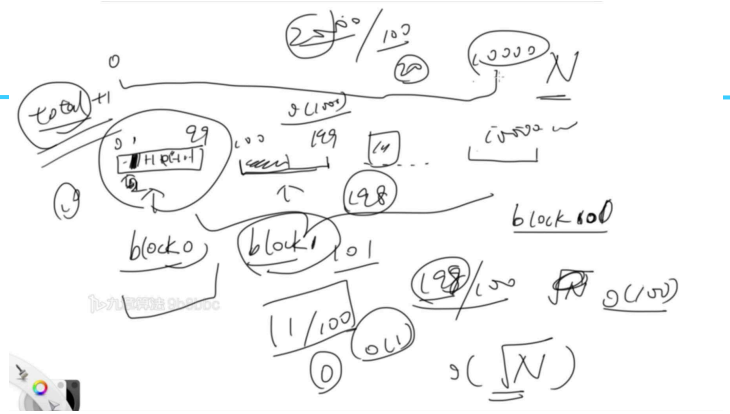


分块检索算法

将长度为 N 的区间分成 √N 的大小的小区间

总共 √N 个小区间，每个小区间统计局部的数据

因此在这些区间中进行增删查改的效率是 O(√N)



统计每个数前面比他小的数

https://www.lintcode.com/problem/count-of-smaller-number-before-itself/

https://www.jiuzhang.com/solution/count-of-smaller-number-before-itself/

[1, 2, 7, 8, 5] 每个数前面比他小的数分别为 [0, 1, 2, 3, 2]

Python 代码

九章算法
www.jiuzhang.com

```python
class BlockArray:
    def __init__(self, max_value):
        self.blocks = [
            Block()
            for _ in range(max_value // 100 + 1)
        ]

    def count_smaller(self, value):
        count = 0
        block_index = value // 100
        for i in range(block_index):
            count += self.blocks[i].total

        counter = self.blocks[block_index].counter
        for val in counter:
            if val < value:
                count += counter[val]
        return count

    def insert(self, value):
        block_index = value // 100
        block = self.blocks[block_index]
        block.total += 1
        block.counter[value] = block.counter.get(value, 0) + 1
```

```python
class Block:
    def __init__(self):
        self.total = 0
        self.counter = {}
```

```python
class Solution:
    """
    @param A: an integer array
    @return: A list of integers includes the inde
    """
    def countOfSmallerNumberII(self, A):
        if not A:
            return []

        block_array = BlockArray(10000)
        results = []
        for a in A:
            count = block_array.count_smaller(a)
            results.append(count)
            block_array.insert(a)
        return results
```

Java 代码

九章算法
www.jiuzhang.com

```java
class BlockArray {
    public Block[] blocks;
    public int blockSize;

    public BlockArray(int capacity) {
        blockSize = (int) Math.sqrt(capacity);
        int blockCount = capacity / blockSize + 1;
        blocks = new Block[blockCount];
        for (int i = 0; i < blockCount; i++) {
            blocks[i] = new Block(blockSize);
        }
    }

    public int countSmaller(int value) {
        int index = value / blockSize;
        int count = 0;
        for (int i = 0; i < index; i++) {
            count += blocks[i].total;
        }

        for (int i = 0; i + index * blockSize < value; i++) {
            count += blocks[index].counter[i];
        }

        return count;
    }

    public void insert(int value) {
        int index = value / blockSize;
        blocks[index].total++;
        blocks[index].counter[value - index * blockSize]++;
    }
}
```

```java
class Block {
    public int total;
    public int[] counter;
    public Block(int blockSize) {
        this.total = 0;
        this.counter = new int[blockSize];
    }
}
```

```java
public class Solution {
    /**
     * @param A: an integer array
     * @return: A list of integers includes the index of t
     */
    public List<Integer> countOfSmallerNumberII(int[] A) {
        List<Integer> results = new ArrayList<>();
        if (A == null || A.length == 0) {
            return results;
        }

        BlockArray blockArray = new BlockArray(10000);
        for (int i = 0; i < A.length; i++) {
            results.add(blockArray.countSmaller(A[i]));
            blockArray.insert(A[i]);
        }

        return results;
    }
}
```