快快编程
kkcoding.net

信奥算法

二进制 位运算

快快编程
kkcoding.net

速度更快

左移k位：乘以2^k

<<

二进制左移1位
右侧补零

1<<1是2, 2<<1是4
3<<1是6, 5<<4是80

11111110000000000011111111000100
<<4
得到
11100000000000111111100010000000

快快编程
kkcoding.net

速度更快

右移k位：除以2^k取整

>>

二进制右移1位
左侧补零

2>>1是1, 4>>1是2
7>>1是3, 11>>2是2

11111110000000000001111111000100
>>4
得到
00001111111000000000000111111100

快快编程
kkcoding.net

优先级

位运算的优先级非常低
建议位运算都使用括号

预测结果

```
4    cout<<(1+1<<1)<<endl;
5    cout<<(1+(1<<1))<<endl;

6    cout<<(2+4>>1)<<endl;
7    cout<<(2+(4>>1))<<endl;
```

快快编程
kkcoding.net

分而治之

分治法

归并排序

快快编程
kkcoding.net

老师
演示

归并排序

merge sort

老师播放视频演示

同学观察排序过程

请同学总结排序步骤和方法

原数组均分为左右两部分

先对每部分单独排序

再合并两部分

分治思想

大事化小
小事化了

快快编程
kkcoding.net

代码 主函数

```
1  #include<iostream>
2  using namespace std;
3  const int N=1000009;
4  int n,x[N],y[N];
5  void msort(int l,int r){
19 int main(){
20     cin>>n;
21     for(int i=0;i<n;i++) cin>>x[i];
22     msort(0,n-1);
23     for(int i=0;i<n;i++) cout<<x[i]<<" ";
24     return 0;
25 }
```

对数组l号到r号
进行归并排序

快快编程
kkcoding.net

代码1

```
5  void msort(int l,int r){
6      if(l==r) return;
7      int mid=(l+r)>>1;
8      msort(l,mid);
9      msort(mid+1,r);
10
11
12
13
14
15
16
17
18 }
```

若此时剩1个元素,不用排序
定位中点mid
左半部分l号到mid号排序
右半部分mid+1号到r号排序

将左右已经排序的两部分
进行合并

代码1

电脑上完成
代码和翻译

```
5  void msort(int l,int r){
6      if(l==r) return;
7      int mid=(l+r)>>1;
8      msort(l,mid);
9      msort(mid+1,r);
10     int i=l,j=mid+1;
11     循环填y数组l号到r号元素{
12         如果左部分用完了,填上右部分当前数
13         否则如果右部分用完了,填上左部分当前数
14         否则如果左部分当前数小于右部分当前数,填上左部分当前数
15         否则填上右部分当前数
16     }
17     for(int k=l;k<=r;k++)x[k]=y[k];
18 }
```

若此时剩1个元素,不用排序

定位中点mid

左半部分l号到mid号排序

右半部分mid+1号到r号排序

准备合并,双游标i和j
为左右两部分的起始位置

y数组l号到r号元素拷贝给x数组

代码2

对比代码1
有啥区别?

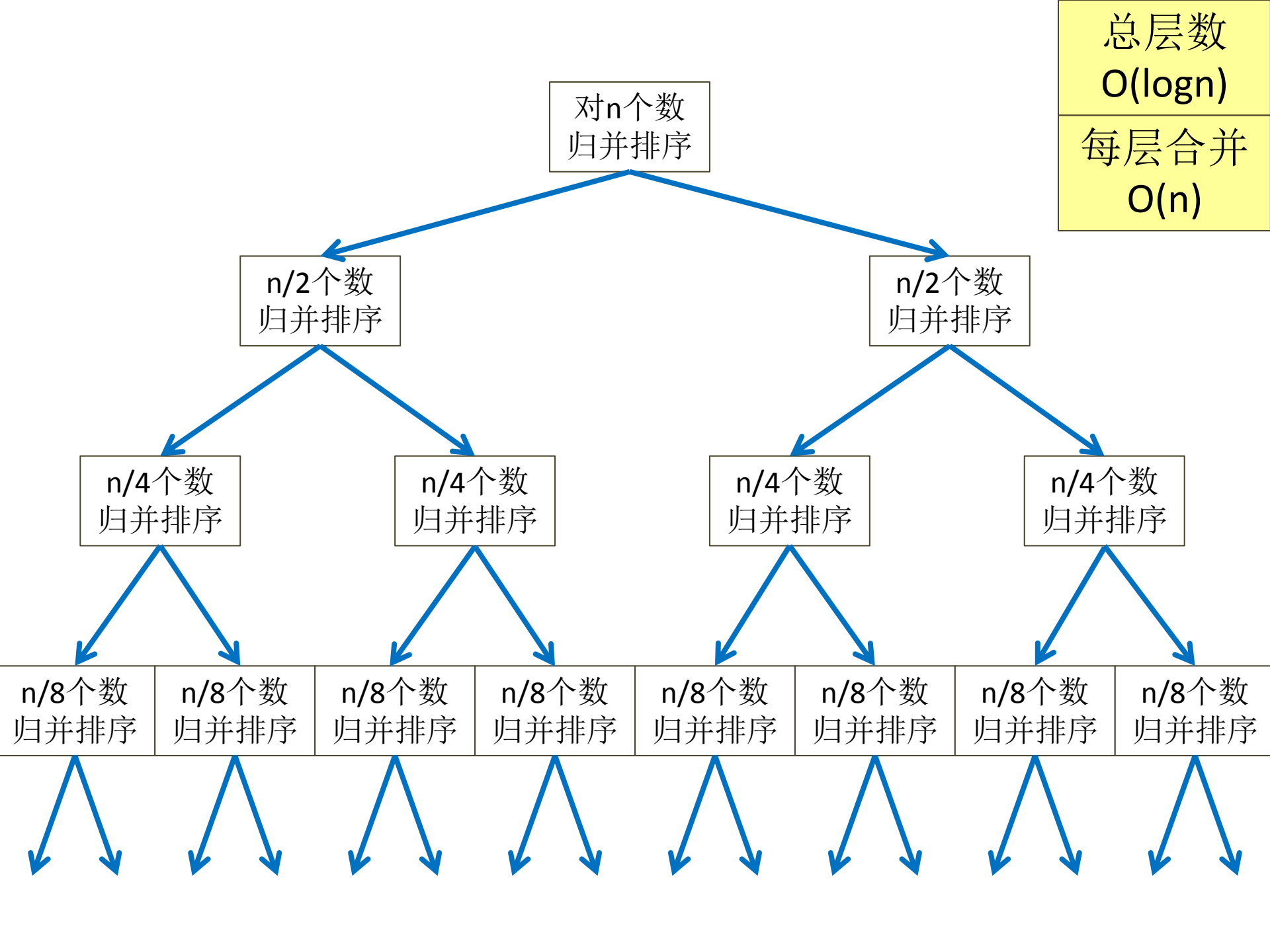
```
5  void msort(int l,int r){
6      if(l==r) return;
7      int mid=(l+r)>>1;
8      msort(l,mid);
9      msort(mid+1,r);
10     int i=l,j=mid+1;
11     for(int k=l;k<=r;k++){
12         if(i>mid) y[k]=x[j++];
13         else if(j>r) y[k]=x[i++];
14         else if(x[i]<=x[j]) y[k]=x[i++];
15         else y[k]=x[j++];
16     }
17     for(int k=l;k<=r;k++)x[k]=y[k];
18 }
```

排序的稳定性

稳定的排序:
当两个数字相同时
不颠倒它们的左右
相对位置

左右部分当前数**相等**时
优先填上左部分当前数

快快编程
kkcoding.net


总层数
O(logn)
每层合并
O(n)
对n个数
归并排序
n/2个数
归并排序
n/2个数
归并排序
n/4个数
归并排序
n/4个数
归并排序
n/4个数
归并排序
n/4个数
归并排序
n/8个数
归并排序
n/8个数
归并排序
n/8个数
归并排序
n/8个数
归并排序
n/8个数
归并排序
n/8个数
归并排序
n/8个数
归并排序
n/8个数
归并排序

时间复杂度O(nlogn)

T(n)表示对n个数字归并排序所需时间

$$
\begin{aligned}
T(n) &= T(n/2)*2+O(n) \\
&= (T(n/4)*2+O(n/2))*2+O(n) \\
&= T(n/4)*4+O(n)*2 \\
&= T(n/8)*8+O(n)*3 \\
&= T(n/16)*16+O(n)*4 \\
&\cdots\cdots \\
&= O(1)*n+O(n)*\log n \\
&= O(n\log n)
\end{aligned}
$$

快快编程 kkcoding.net

小讨论

该算法缺点是什么？

空间需要额外增加
一个数组辅助排序

快快编程
kkcoding.net

代码3

```
5  void msort(int l,int r){
6      if(l==r)return;
7      int mid=(l+r)>>1;
8      msort(l,mid);
9      msort(mid+1,r);
10     int i=l,j=mid+1;
11     for(int k=l;k<=r;k++)
12         if(i>mid||j<=r&&x[i]>x[j]) y[k]=x[j++];
13         else y[k]=x[i++];
14     for(int k=l;k<=r;k++)x[k]=y[k];
15 }
```

请翻译12,13行

这个排序代码是稳定排序吗？

代码4

```
5  void msort(int l,int r){
6      if(l==r)return;
7      int mid=(l+r)>>1;
8      msort(l,mid);
9      msort(mid+1,r);
10     int i=l,j=mid+1,k=l;
11     while(i<=mid||j<=r){
12         if(i>mid||j<=r&&x[i]>x[j]) y[k++]=x[j++];
13         else y[k++]=x[i++];
14     }
15     for(int k=l;k<=r;k++)x[k]=y[k];
16 }
```

请翻译10-13行

这个排序代码是稳定排序吗？

课间
任务

请用电脑
挑选代码3或代码4
完成归并排序
完整程序
限时10分钟

快快编程
kkcoding.net

逆序对
计数

快快编程
kkcoding.net

求逆序对

给定一个序列a1,a2,…,an，如果存在i<j但ai>aj，那么我们称(i,j)为逆序对，求逆序对的数目。

输入第一行为n,表示序列长度，接下来有n行，第i+1行表示序列中的第i个数。
n<=100000，ai<=100000
输出一个整数代表所有逆序对总数

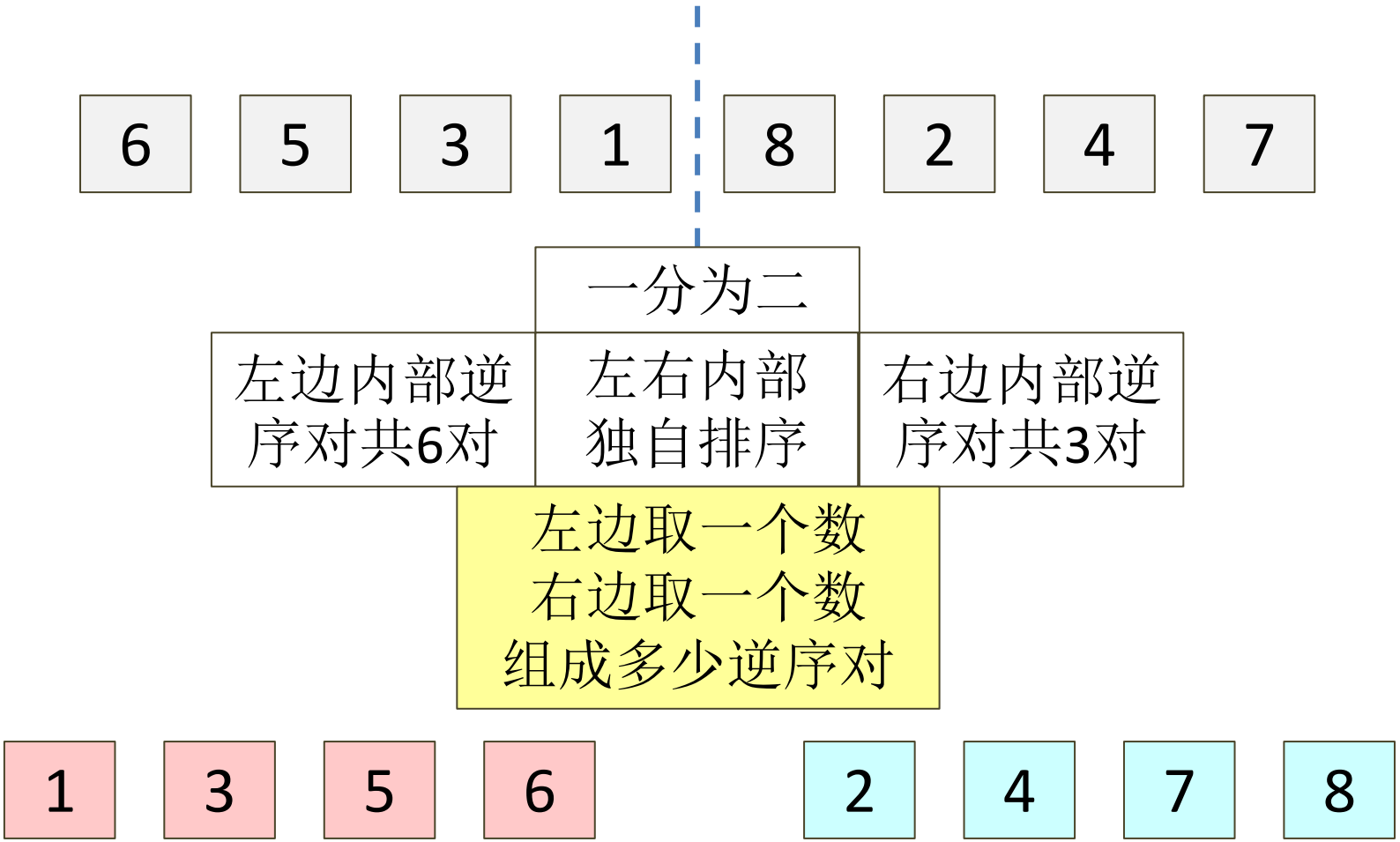
输入样例	输入样例	输入样例
3 3 2 1	5 1 2 1 0 3	6 3 2 1 3 2 1
输出样例	输出样例	输出样例
3	4	9

计数问题

暴力枚举
该怎么做?

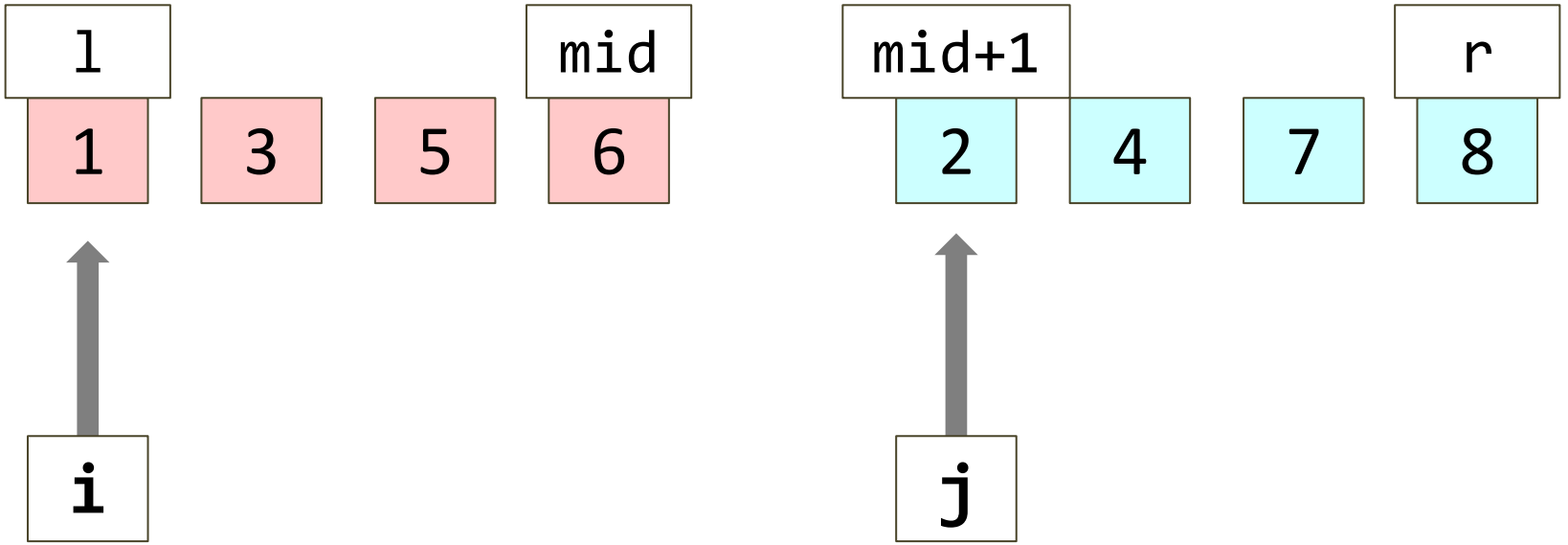
借鉴归并排序

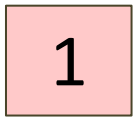
分治法
该怎么做?



快快编程
kkcoding.net

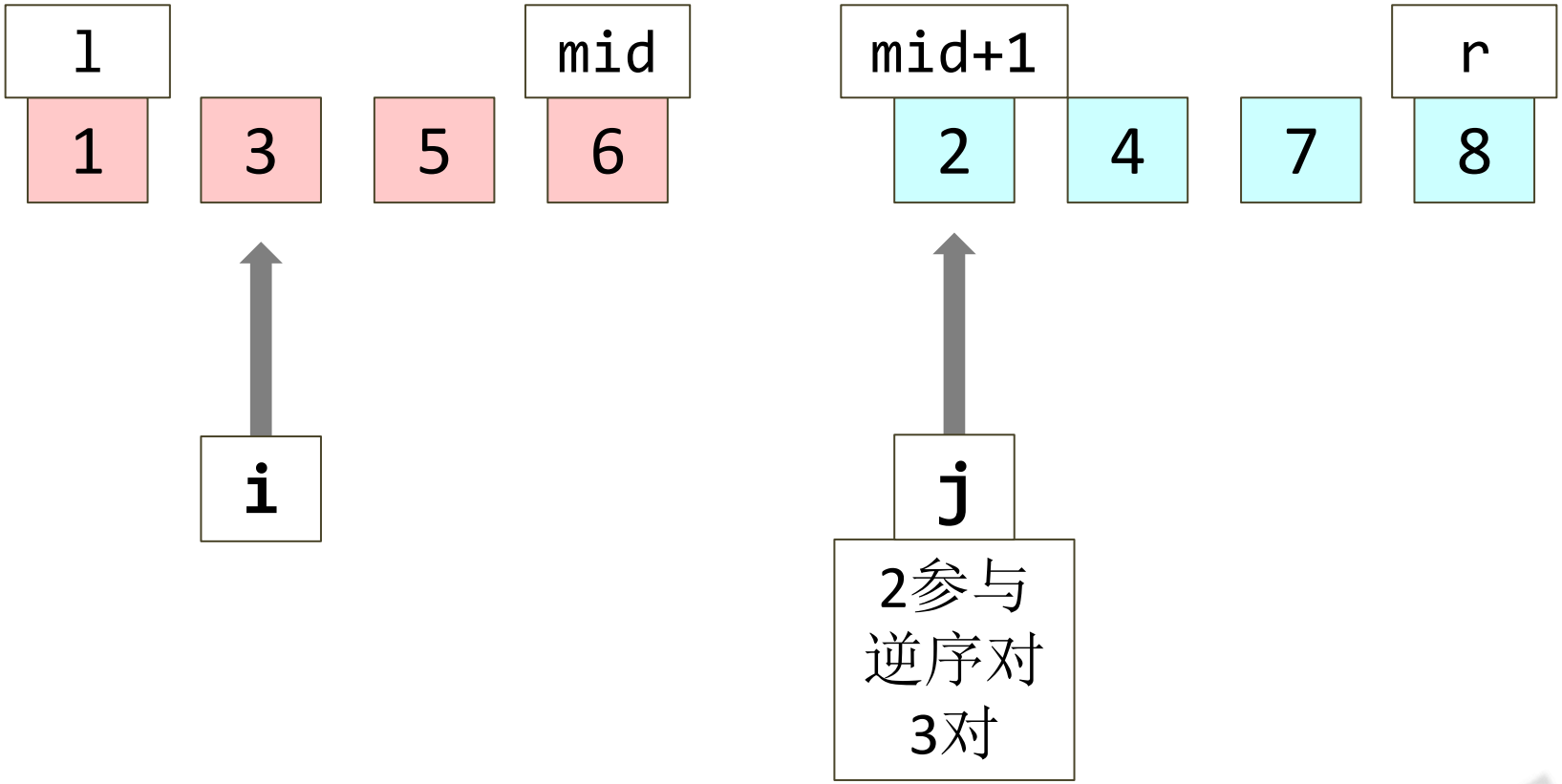
左边取一个数,右边取一个数,组成多少逆序对





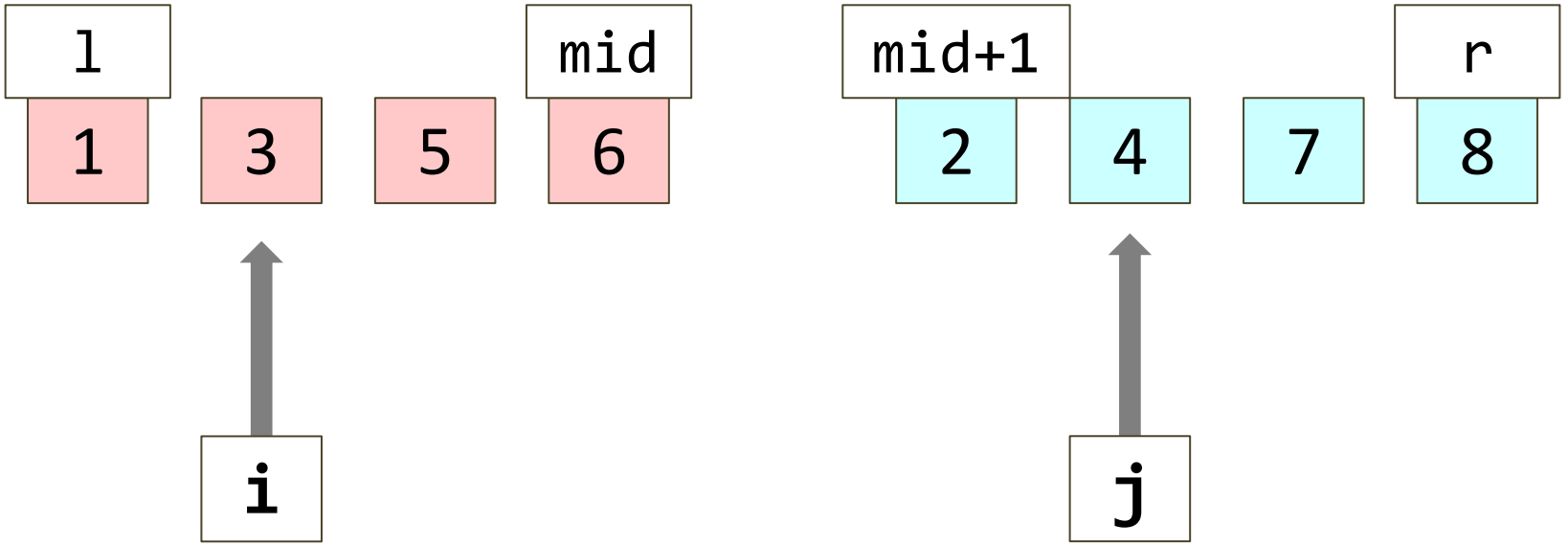
快快编程
kkcoding.net

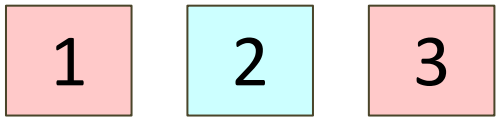
左边取一个数,右边取一个数,组成多少逆序对



快快编程
kkcoding.net

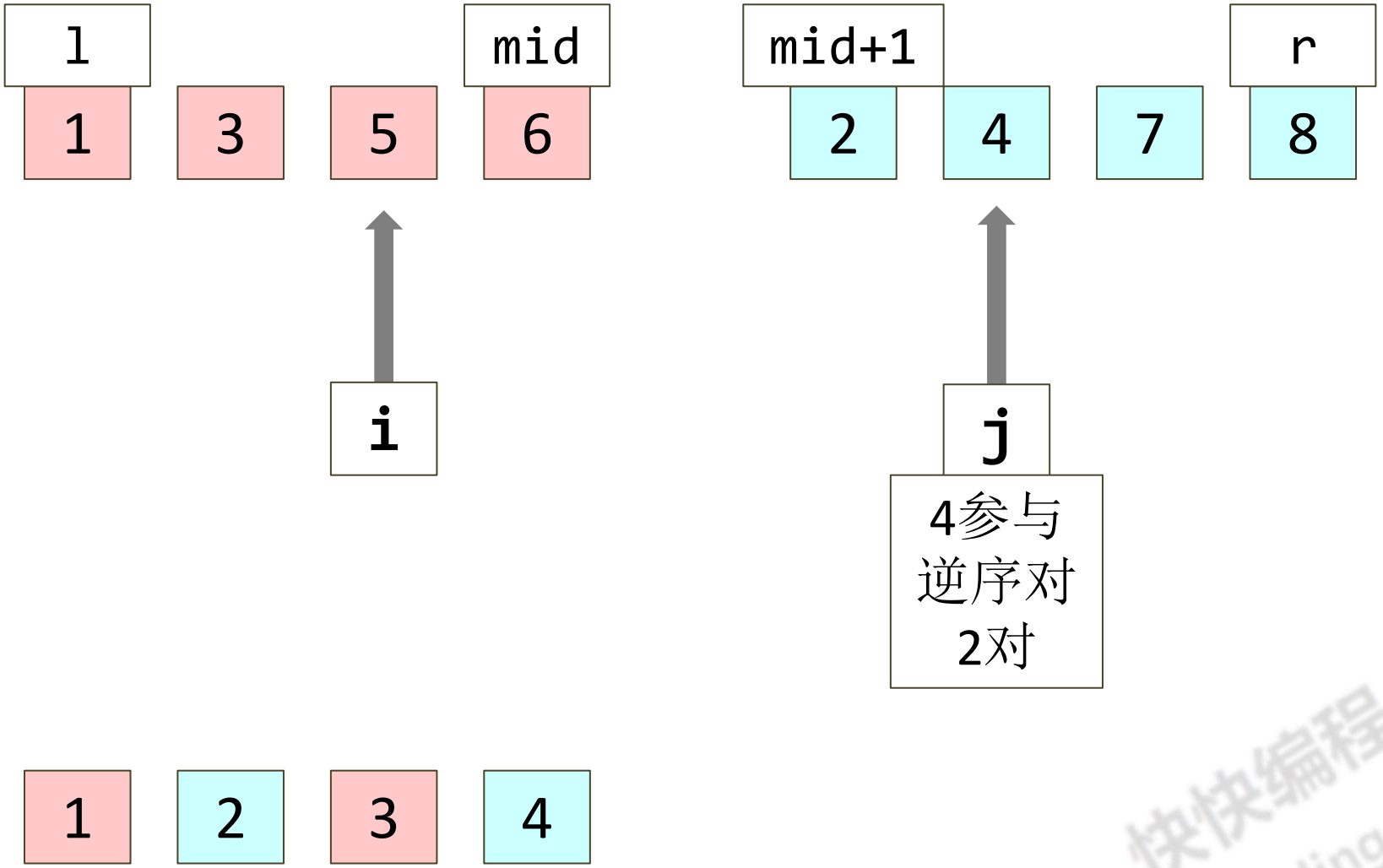
左边取一个数,右边取一个数,组成多少逆序对





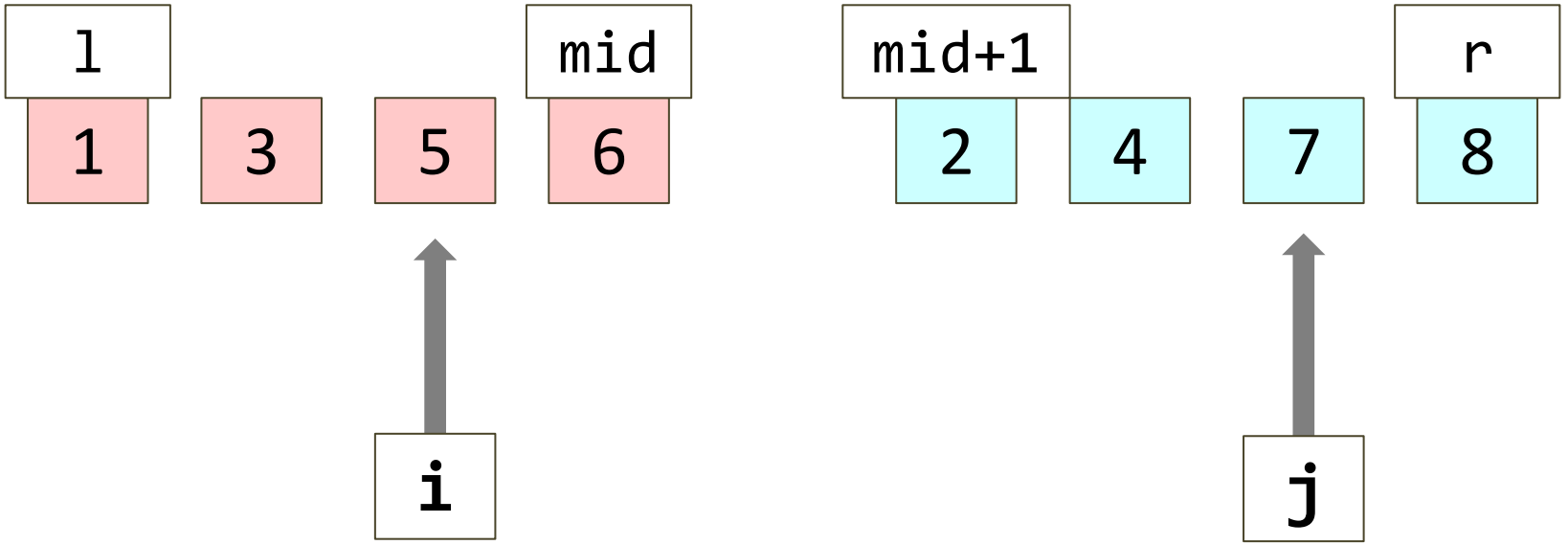
快快编程
kkcoding.net

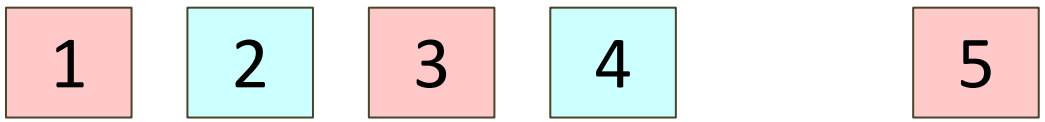
左边取一个数,右边取一个数,组成多少逆序对



快快编程
kkcoding.net

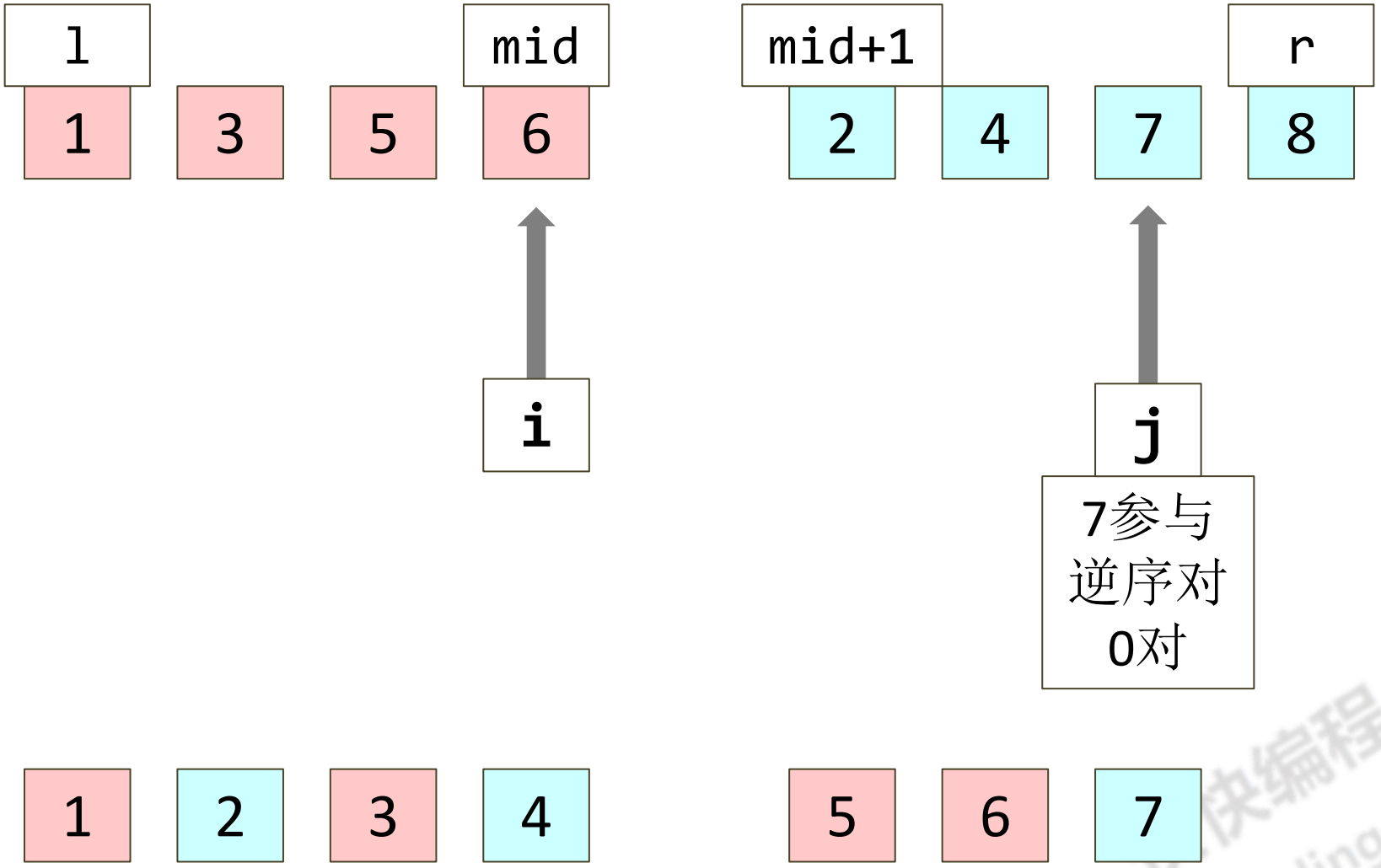
左边取一个数,右边取一个数,组成多少逆序对





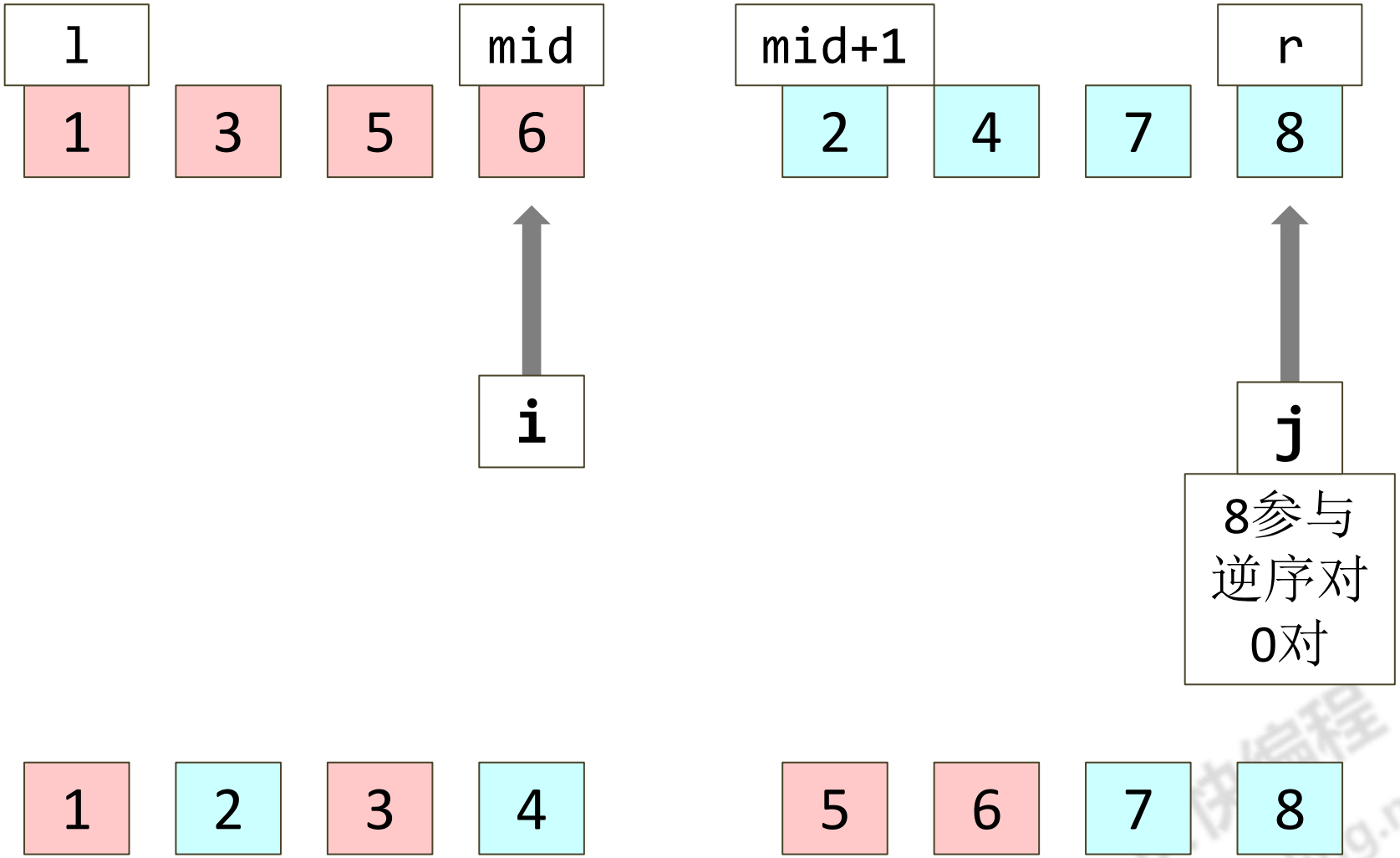
快快编程
kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



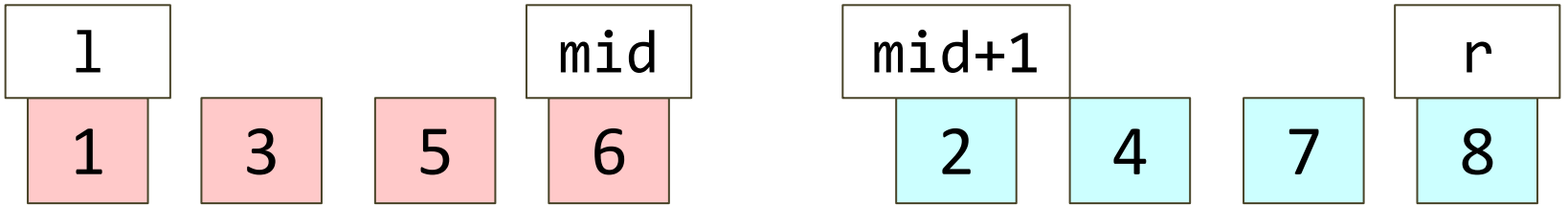
快编程
kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



3	5	6	3对	2
	5	6	2对	4
			0对	7
			0对	8

右边数确定后左边可以搭配几个数?

右边取一个数参与逆序对

计数问题

加法原理
分类讨论

按右边数分类

快快编程
kkcoding.net

代码1

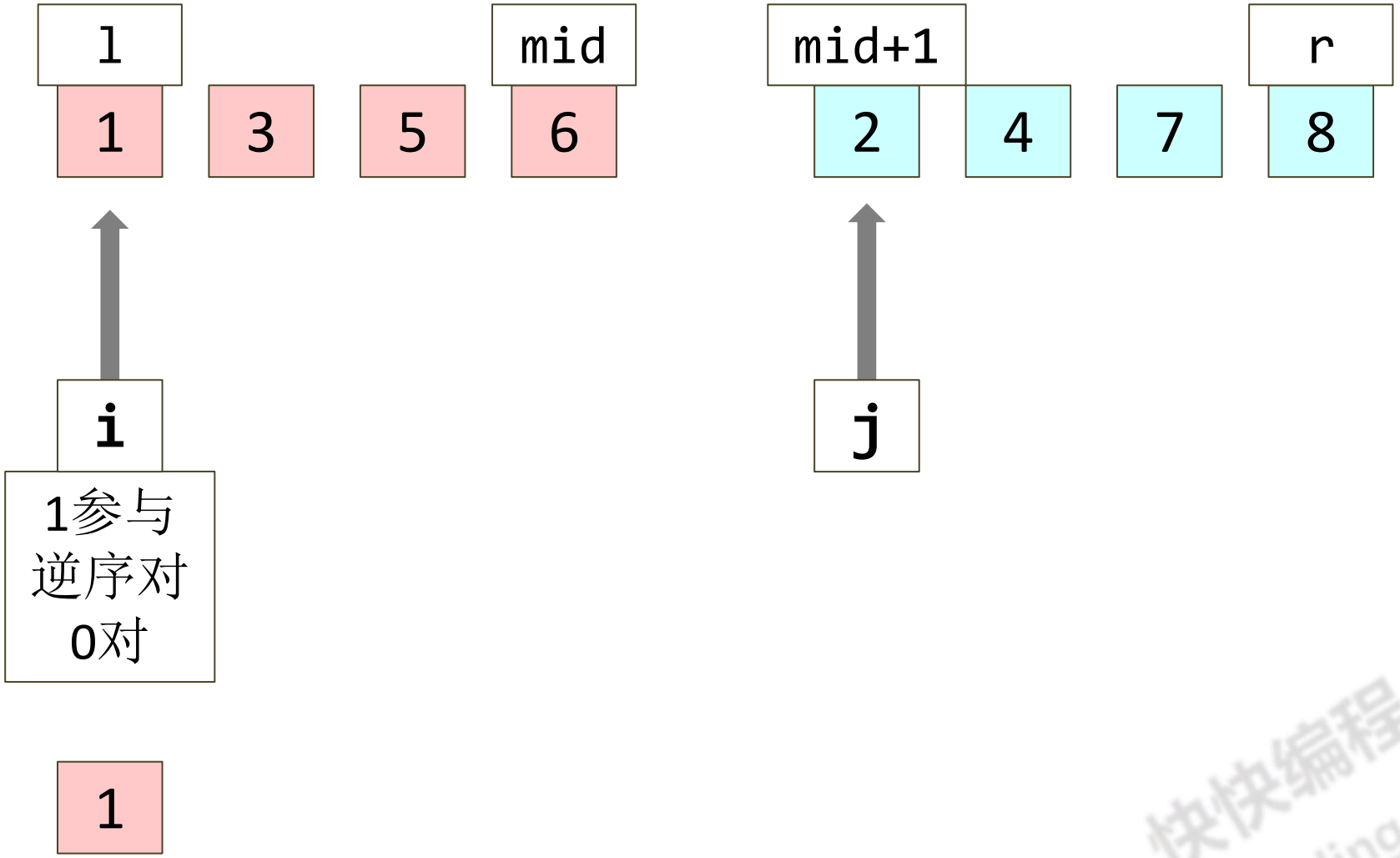
```
6  ll solve(ll l,ll r){
7      if(l==r)return 0;
8      ll mid=(l+r)>>1;
9      ll ret=0;
10     ret+=solve(l,mid);
11     ret+=solve(mid+1,r);
12     ll i=l,j=mid+1;
13     for(ll k=l;k<=r;k++){
14         if(i>mid) y[k]=x[j++];
15         else if(j>r) y[k]=x[i++];
16         else if(x[i]<=x[j]) y[k]=x[i++];
17         else {ret+=mid-i+1;y[k]=x[j++];}
18     }
19     for(ll k=l;k<=r;k++)x[k]=y[k];
20     return ret;
21 }
```

求出数组l号到r号内
有多少逆序对

轮流翻译每一行

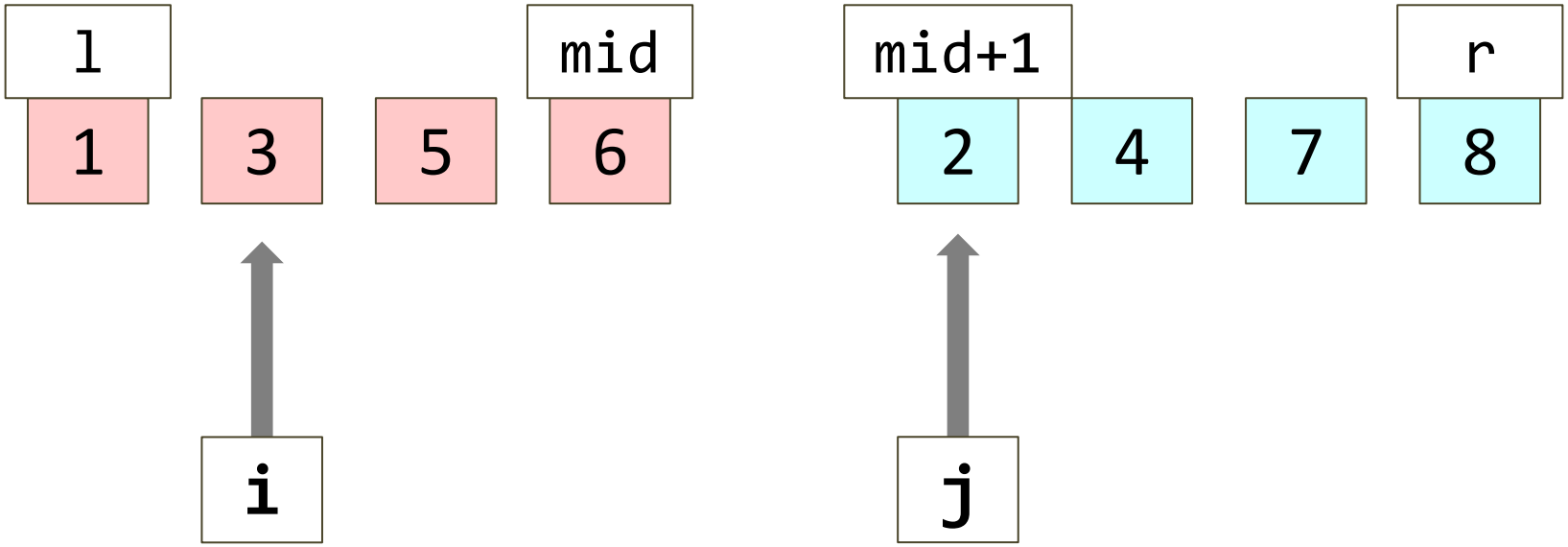
快快编程
kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



快快编程
kkcoding.net

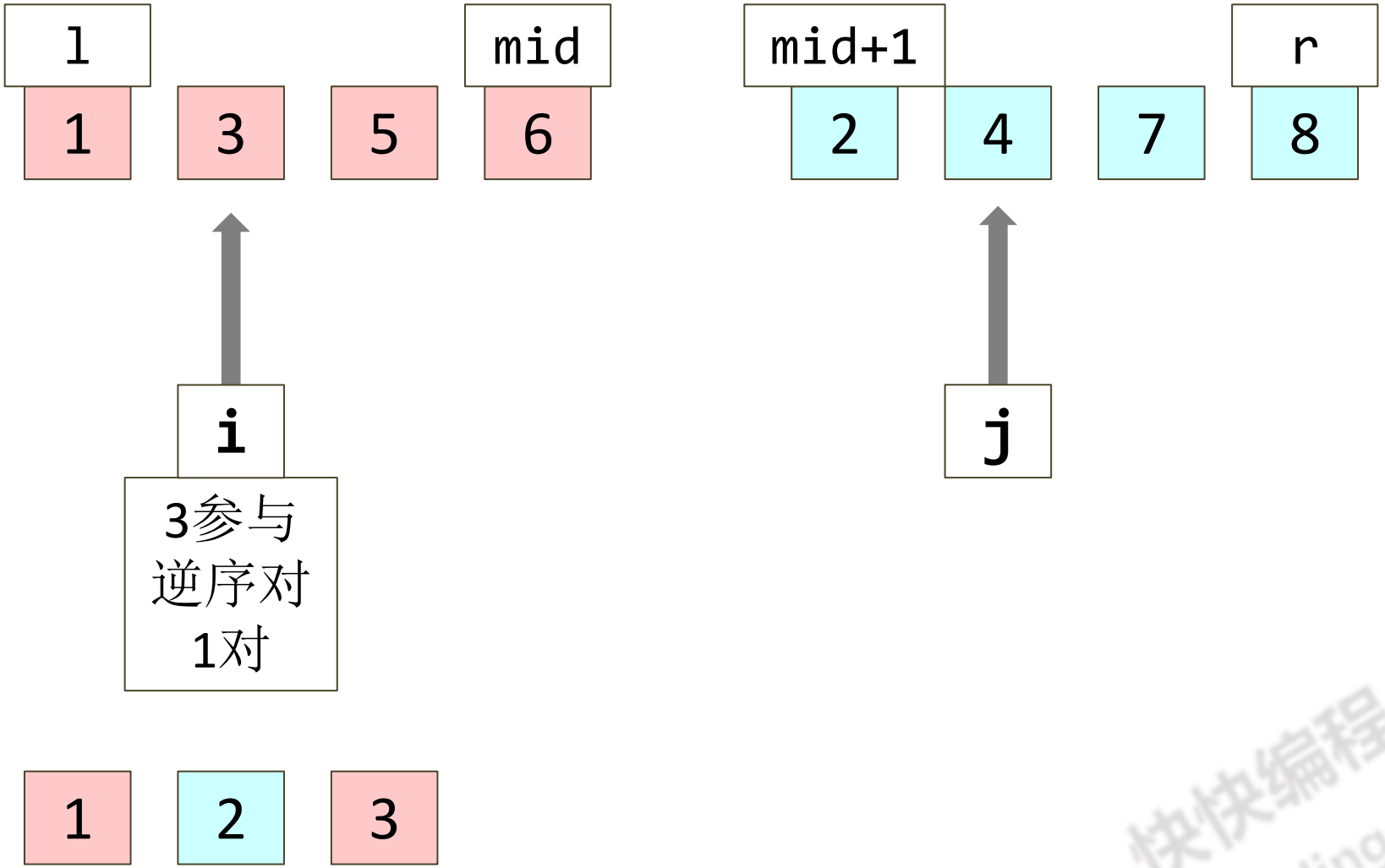
左边取一个数,右边取一个数,组成多少逆序对



1

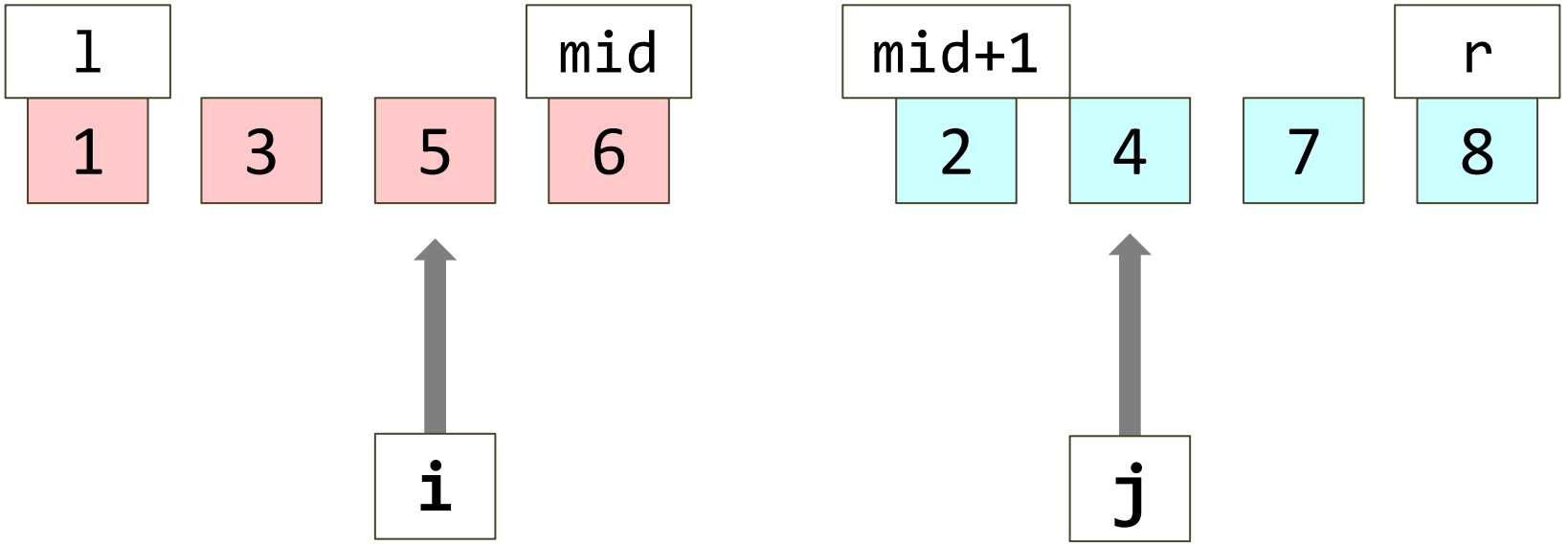
快快编程
kkcoding.net

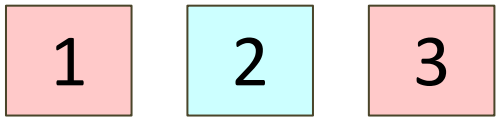
左边取一个数,右边取一个数,组成多少逆序对



快快编程
kkcoding.net

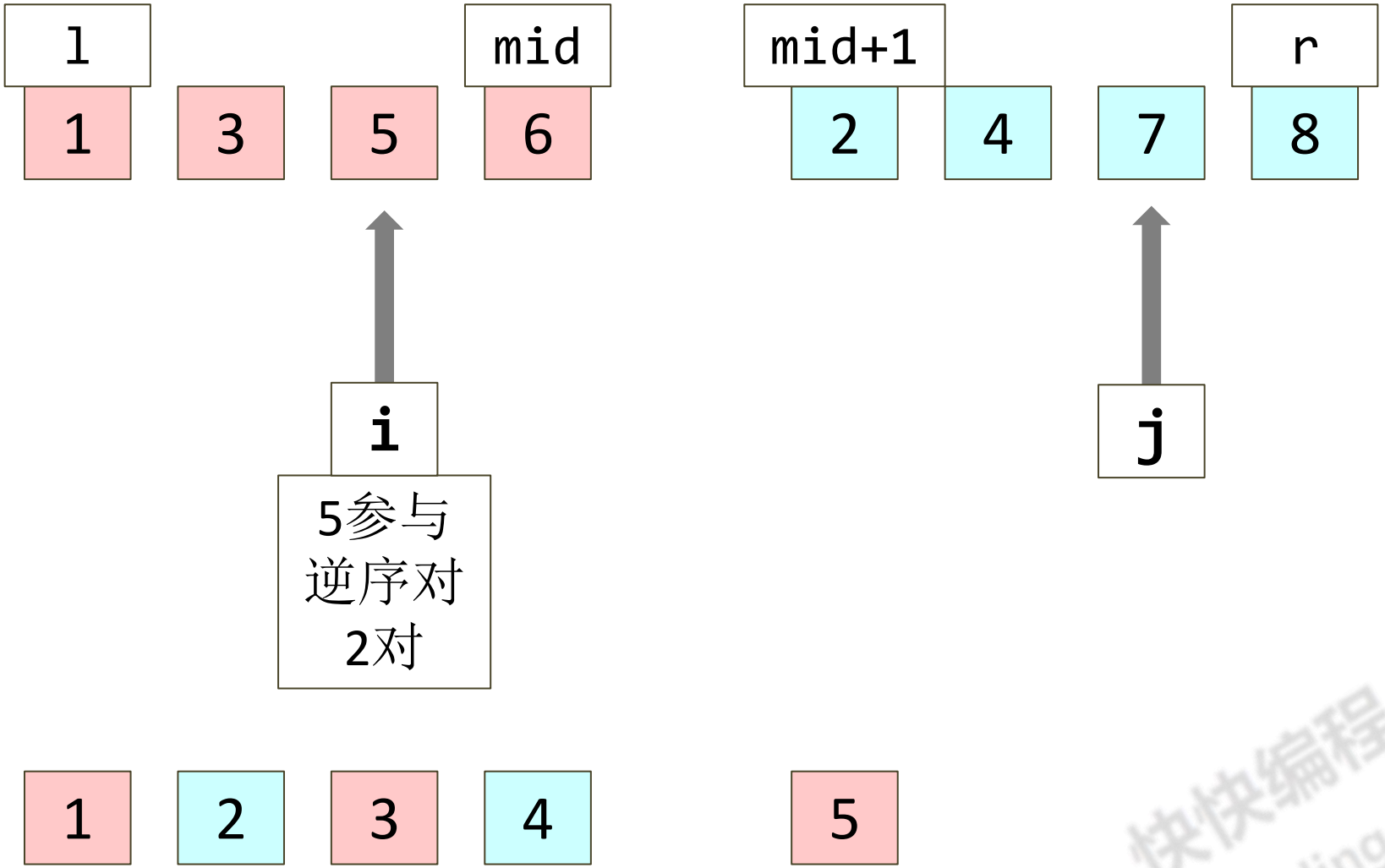
左边取一个数,右边取一个数,组成多少逆序对





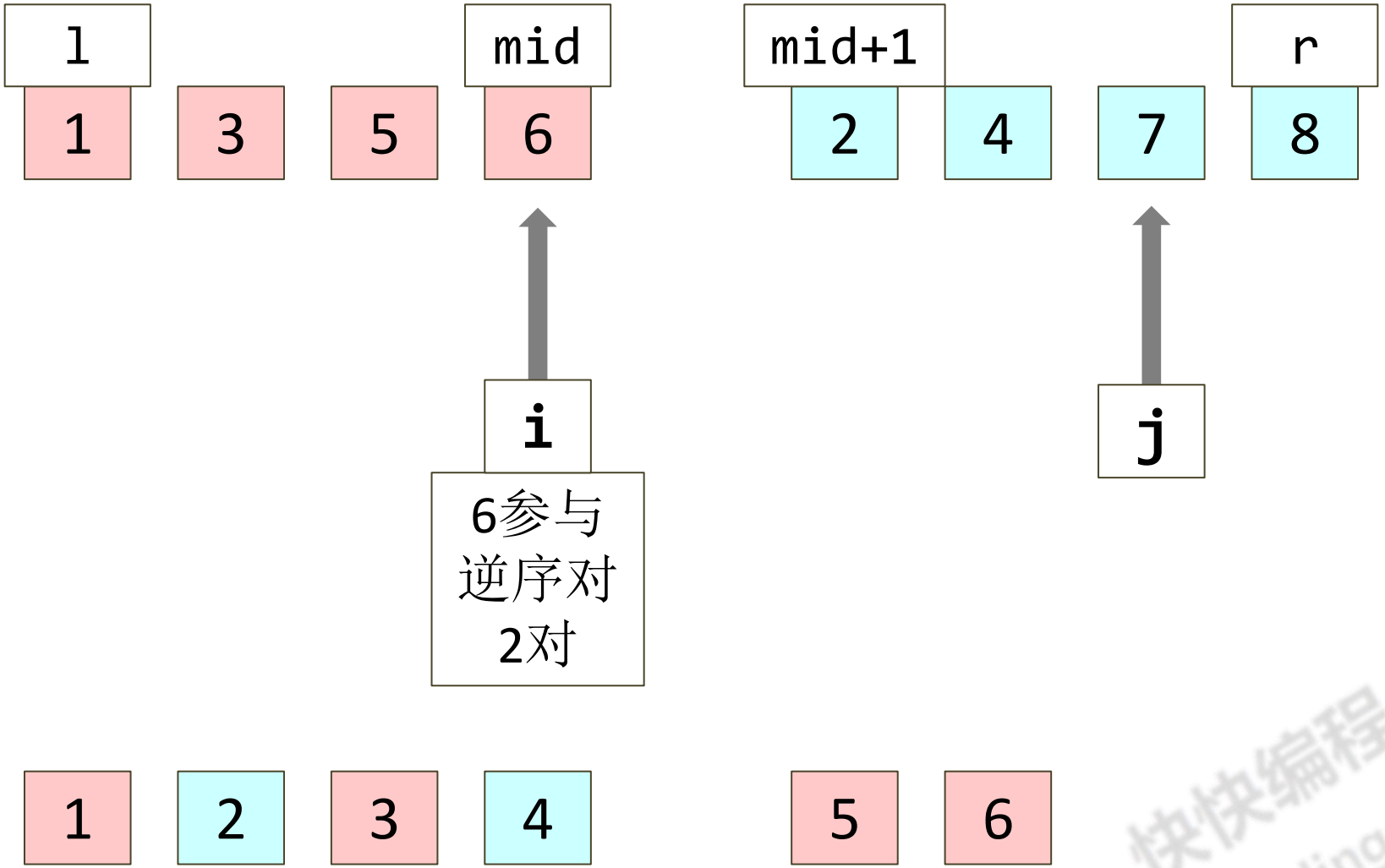
快快编程
kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



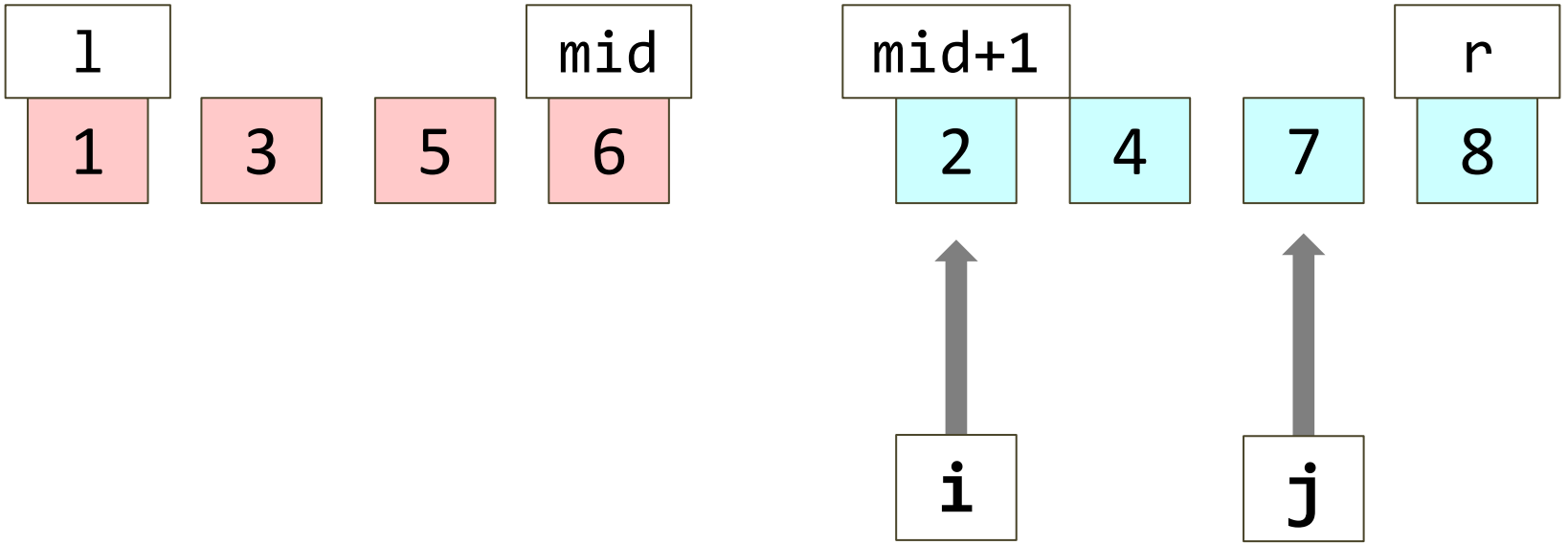
快快编程
kkcoding.net

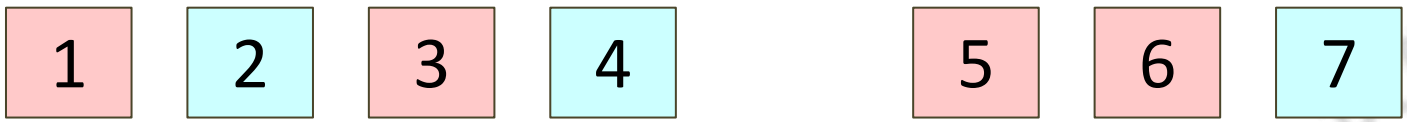
左边取一个数,右边取一个数,组成多少逆序对



快快编程
kkcoding.net

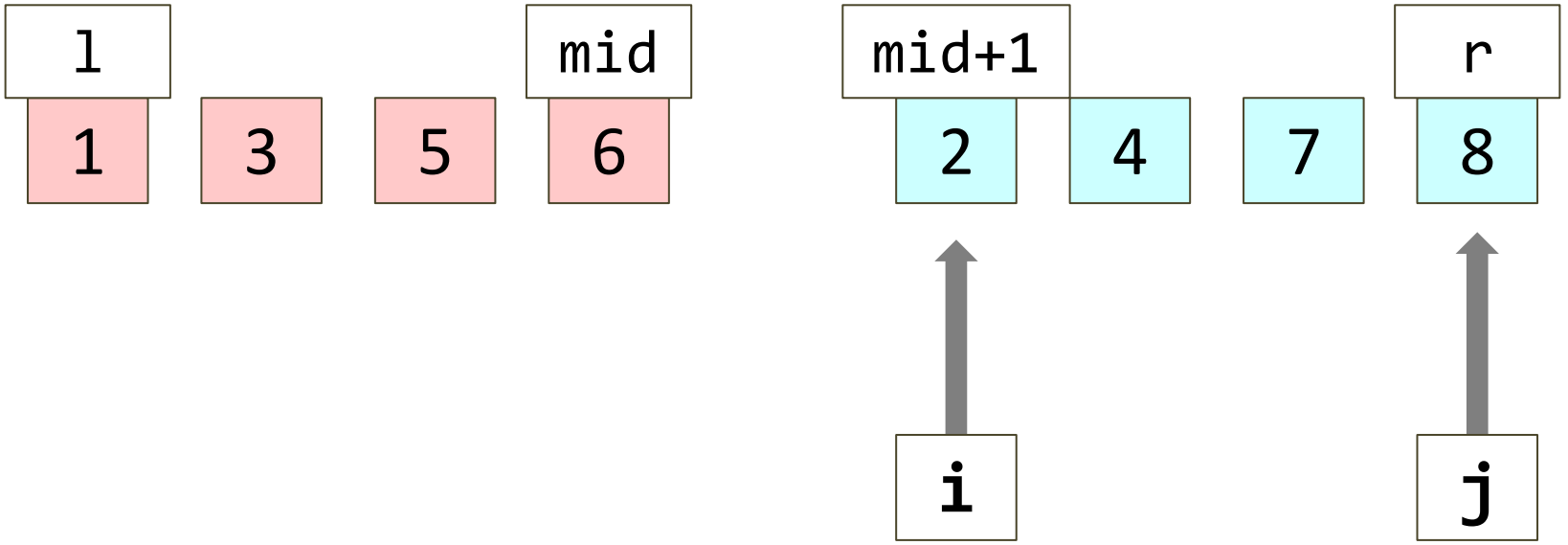
左边取一个数,右边取一个数,组成多少逆序对





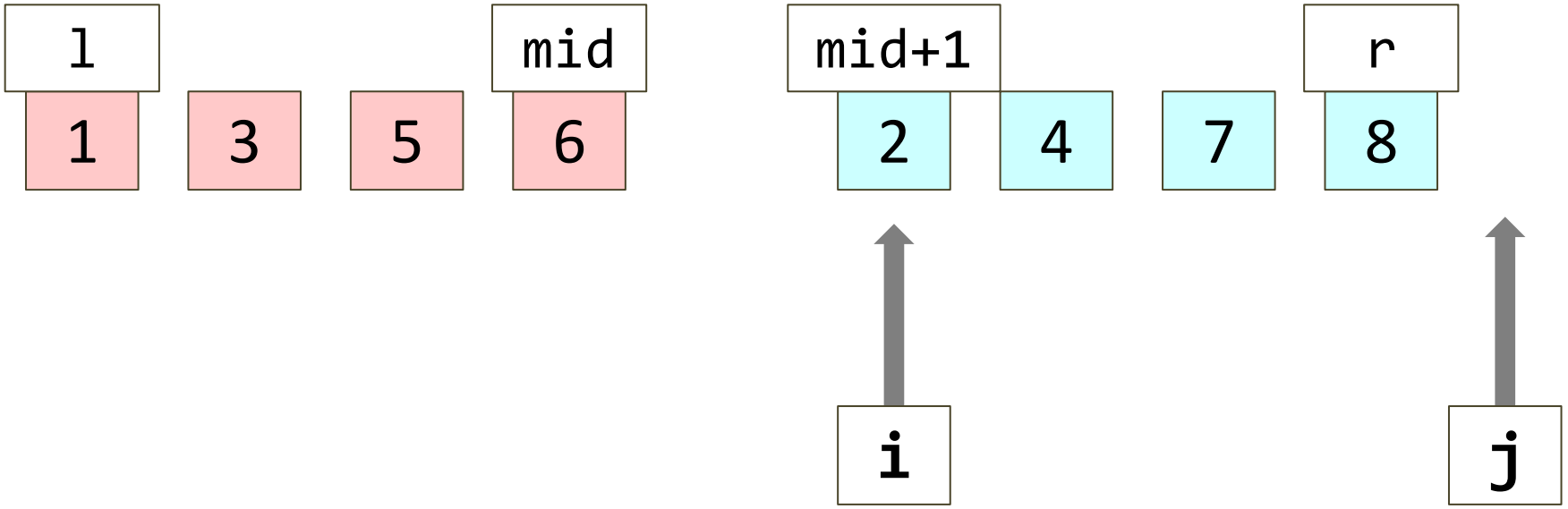
快编程
kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



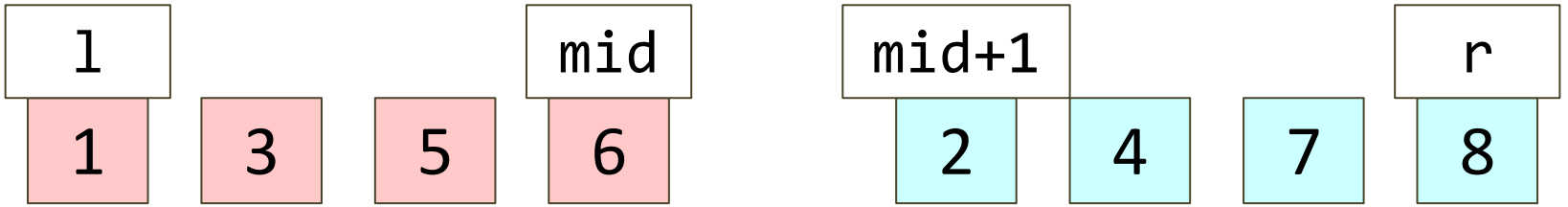
kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



kkcoding.net

左边取一个数,右边取一个数,组成多少逆序对



1	0对		
3	1对	2	
5	2对	2	4
6	2对	2	4

左边取一个数
参与逆序对

左边数确定后
右边可以搭配
几个数?

计数问题

加法原理
分类讨论

按左边数
分类

快快编程
kkcoding.net

代码2

```
6  ll solve(ll l,ll r){
7      if(l==r)return 0;
8      ll mid=(l+r)>>1;
9      ll ret=0;
10     ret+=solve(l,mid);
11     ret+=solve(mid+1,r);
12     ll i=l,j=mid+1;
13     for(ll k=l;k<=r;k++){
14         if(i>mid) y[k]=x[j++];
15         else if(j>r) {ret+=r-mid; y[k]=x[i++];}
16         else if(x[i]<=x[j]) {ret+=j-mid-1;y[k]=x[i++];}
17         else y[k]=x[j++];
18     }
19     for(ll k=l;k<=r;k++)x[k]=y[k];
20     return ret;
21 }
```

理解15-16行

快快编程 kkcoding.net

代码3

```
6  ll solve(ll l,ll r){
7      if(l==r)return 0;
8      ll mid=(l+r)>>1;
9      ll ret=0;
10     ret+=solve(l,mid);
11     ret+=solve(mid+1,r);
12     ll i=l,j=mid+1;
13     for(ll k=l;k<=r;k++)
14         if(i>mid||j<=r&&x[i]>x[j]) {
15             ret+=mid-i+1;
16             y[k]=x[j++];
17         }
18         else y[k]=x[i++];
19     for(ll k=l;k<=r;k++)x[k]=y[k];
20     return ret;
21 }
```

理解14-16行

快快编程 kkcoding.net

代码4

```
 6  ll solve(ll l,ll r){
 7      if(l==r)return 0;
 8      ll mid=(l+r)>>1;
 9      ll ret=0;
10      ret+=solve(l,mid);
11      ret+=solve(mid+1,r);
12      ll i=l,j=mid+1;
13      for(ll k=l;k<=r;k++)
14          if(i>mid||j<=r&&x[i]>x[j])
15              y[k]=x[j++];
16          else {
17              ret+=j-mid-1;
18              y[k]=x[i++];
19          }
20      for(ll k=l;k<=r;k++)x[k]=y[k];
21      return ret;
22  }
```

理解16-18行

快快编程 kkcoding.net

代码5 错误版

```
6  ll solve(ll l,ll r){
7      if(l==r)return 0;
8      ll mid=(l+r)>>1;
9      ll ret=0;
10     ret+=solve(l,mid);
11     ret+=solve(mid+1,r);
12     ll i=l,j=mid+1;
13     for(ll k=l;k<=r;k++)
14         if(i>mid||j<=r&&x[i]>=x[j])
15             y[k]=x[j++];
16         else {
17             ret+=j-mid-1;
18             y[k]=x[i++];
19         }
20     for(ll k=l;k<=r;k++)x[k]=y[k];
21     return ret;
22 }
```

错在哪里?
请找最简反例

输入
2
5 5
输出1

快快编程 kkcoding.net

快快编程
kkcoding.net

快快编程作业

969	使用归并排序
122	

快快编程
kkcoding.net