专题一：基础算法

陈市

天津大学

2023 年 6 月 26 日

目录

目录

目录

输入方法

例如：题目要求输入两个数，求两个数的和。

单组样例

- 输入描述：给两个数，a 和 b
- 示例输入：
  1 2

```
cin >> a >> b;
cout << a + b << endl;

scanf("%d%d",&a,&b);
printf("%d\n",a+b);
```

输入方法

例如：题目要求输入两个数，求两个数的和。

多组样例-要求输入测试样例数量

- 输入描述：有多组测试样例，首先输入测试样例的数量，再输入每组样例中的两个数 a 和 b
- 示例输入：
  2
  1 2
  3 4

```
1 cin >> T;
2 for (int i = 0; i < T; i++) {
3     cin >> a >> b;
4     cout << a + b << endl;
5 }
```

输入方法

例如：题目要求输入两个数，求两个数的和。

多组样例—给出输入终止条件

- 多组测试样例，每个样例有两个整数 a 和 b，当 a 为-1 并且 b 为-1 的时候终止输入
- 示例输入：
  1 2
  2 3
  -1 -1

```
while(cin >> a >> b) {
    if (a == -1 && b == -1)break;
    cout << a + b << endl;
}
```

输入方法

例如：题目要求输入两个数，求两个数的和。

多组样例—输入到文件尾结束

- 输入描述：有多组测试样例，每组样例有两个数 a 和 b
- 示例输入：
  1 2
  3 4

```
while(cin >> a >> b) {
    cout << a + b << endl;
}

while(scanf( "%d%d" ,&a,&b) != EOF) {
    cout << a + b << endl;
}
```

目录

举个例子

Description

求最长的回文串。

Solve

对应这个问题我们可以有很多的方法
枚举左右端点，然后判断该串是否为回文串，复杂度 $T(n) = O(n^3)$

```
1 bool check(int l,int r){
2   while(l<r&&s[l]==s[r]) l++,r--;
3   return l>=r;
4 }
5 for(int i=1;i<=n;i++)
6   for(int j=i;j<=n;j++)
7     if(check(i,j)) ans=max(ans,j-i+1);
```

举个例子

Description

求最长的回文串。

Solve

对应这个问题我们可以有很多的方法
枚举中心点，然后向两侧扩展至最大长度，复杂度 $T(n) = O(n^2)$

```
1 for(int i=1;i<=n;i++){
2   int l=i,r=i;
3   while(s[l]==s[r]) l--,r++;
4   ans=max(ans,r-l+1);
5 }
```

当然我们还有更快的方法，但是这不是我们要讨论的内容。

如何判定自己的程序跑的更快

1. 通过上面两个例子，我们可以看出不同算法间具有不同的时间消耗。
2. 第二个算法中 $T(n) = 2(\sum_{i=1}^{n/2} i + \sum_{i=n/2}^{1} i) = \frac{1}{2}n^2 + n = O(n^2)$
3. $\frac{1}{2}n^2 + n$ 不是与 n^2 相等，而是同阶
4. 同阶可以简单的理解成同一个数量级，我们不需要准确的计算出计算次数，只需要知道大概的数量级就可以完成我们的算法设计了。

表: 不同运算量对应的数据范围

运算量	$n!$	2^n	n^3	n^2	$nlogn$	$n\sqrt{n}$	n	$logn$
最大规模	11	25	300	5000	$5 * 10^5$	$5 * 10^4$	10^8	10^{18}

目录

枚举法

- 枚举算法是在程序设计中最为常见的一个算法，其核心在于枚举所有的可能性，找到可行解。枚举算法通常是将问题可能的答案一一列举，根据问题要求判断答案是否可行。保留可行解，舍弃不可行解，最终得出答案。
- 在实际生活中，枚举法十分常见。例如，询问从公元元年到今年经过了几个闰年，就可以枚举 2023 个年份依次判断;

例题

题目 Codeforces557A

- 有一个 n 行 n 列的表。表中第 i 行第 j 列的元素的值为 $i \times j$，且行和列的序号从 1 开始。现在给出一个正整数 x，你的任务是计算该数 x 在表中出现的次数。

例题

题目 Codeforces557A

- 有一个 n 行 n 列的表。表中第 i 行第 j 列的元素的值为 i × j，且行和列的序号从 1 开始。现在给出一个正整数 x，你的任务是计算该数 x 在表中出现的次数。

解法

- 考虑到乘法表上某一行的每个数必定不同，故一行只可能最多出现一次 x，且 x 一定是该行的行号的倍数。
- 于是可以枚举每一行，计算该行的行号 i 是否为 x 的因数，且其列数 xi 小于等于 n。如此一来时间复杂度降低为 $O(n)$。

例题

题目 hduoj1172

- 计算机随机产生一个四位数，玩家每次猜一个数，计算机会告诉他猜对了几个数字，以及几个数字在正确的位置。现在有多组数组，给你每组对话过程，让你确定这个四位数是多少。总对话次数 N<=100。
- 例如随机产生的数字为 1122，玩家猜 1234，计算机会告诉你有 2 个数字是正确的，有 1 个数在正确的位置上。

例题

题目 hduoj1172

- 计算机随机产生一个四位数，玩家每次猜一个数，计算机会告诉他猜对了几个数字，以及几个数字在正确的位置。现在有多组数组，给你每组对话过程，让你确定这个四位数是多少。总对话次数 N<=100。
- 例如随机产生的数字为 1122，玩家猜 1234，计算机会告诉你有 2 个数字是正确的，有 1 个数在正确的位置上。

解法

- 四位数总共也不到 1e4，而数据也只有 100 组，所以可以采用枚举的方法。对每个枚举的四位数进行 N 次判断，如果这个数全部符合，则它就有可能是计算机产生的那个数。如果有多个这样的数，显然不能确定。

目录

概念

模拟就是用计算机来模拟题目中要求的操作，按照题目给出的十分明确的规则对输入数据处理，并按照输出规则输出。这类题目通常需要仔细阅读题面，标注出关键语句，避免出现信息遗漏。思考所有需要注意的细节与方面，编写代码时代码层次分明，并加上适当的注释辅助阅读代码。

特点

模拟题有着非常鲜明的三点特征:

- 题目描述较长，主要是为了清晰地描述题目定义的规则 (也有题目因使用公认的规则而描述较短，如大数的四则运算等)。
- 基本不涉及高深的算法，也不需要对题目本身进行较深入的思考。
- 代码量大，细节较多，出现错误不易排查。

目录

贪心法

贪心法

贪婪算法，又称贪心算法。是指从问题的初始状态出发，通过若干次的贪心选择当前最优值来期望得到全局最优值的一种解题方法。

从“贪心”一词可以看出，贪心策略总是做出在当前看来是最优的选择，也就是说贪心策略并不是从整体上加以考虑，它所做出的选择只是在某种意义下的局部最优解，而许多问题自身的特性决定了该题运用贪心策略可以得到最优解或较优解。

例题

题目 POJ2287

- 田忌赛马加强版，田忌和齐王各 n 匹马，知道这 2n 匹马的速度，问田忌最多可以赢几局？

解法

- 上马对中马，中马对下马这种策略本身就是一种贪心策略，这道题稍为复杂一点，要充分利用每匹马的战斗力。
- 每匹马要尽量战胜对方速度大的马（实力接近，略胜一筹），完全无法获胜的马去消耗对手的快马。

例题

题目

- 有 10 块宝石，每块宝石的体积是 1，价值不等，分别为 1，2，3，4，5，6，7，8，9，10，有一个体积为 5 的背包，怎样选择宝石，使价值最大

解法

- 贪心策略：每次取价值最大的宝石
- 每块宝石的体积也不等，分别为 1，2，3，4，5，6，7，8，9，10，怎样选择？
- 宝石的体积分别为 1，1，1，1，5，6，7，8，9，10 呢？
- 贪心的适用范围有限

目录

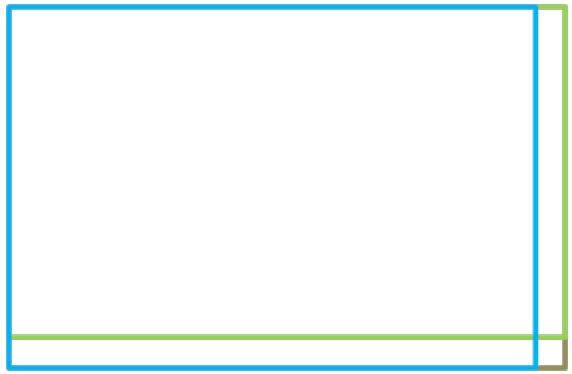
什么是前缀和？

Description

前缀和是一种重要的预处理，能大大降低查询的时间复杂度假设数列：$a_0, a_1, a_2, a_3 \cdots$，则有前缀和 $s_0, s_1, s_2, s_3 \cdots$ 满足：

- $s_0 = a_0$
- $s_1 = a_0 + a_1$
- $s_2 = a_0 + a_1 + a_2$
- $s_n = a_0 + a_1 + a_2 + \cdots + a_n$

二维前缀和



- $s_{i,j} = s_{i-1,j} + s_{i,j-1} - s_{i-1,j-1} + a_{i,j}$

什么是差分？

Description

假设数列：$a_0, a_1, a_2, a_3 \cdots$，则有差分数列 $b_1, b_2, b_3 \cdots$ 满足：

- $b_1 = a_1 - a_0$
- $b_2 = a_2 - a_1$
- $b_n = a_n - a_{n-1}$

解决了什么问题？

- 操作一：将数组 a 中 [L, R] 位置的元素都加上 P
- 操作二：将数组 a 中 [L, R] 位置的元素都减去 P
- 最后询问各元素值

例题

题目 Luogu P2280

- 地图上有 N 个目标，用整数 X,Y 表示其位置，每个目标有一个价值，现投放激光炸弹，其爆破范围是一个边长为 R 的正方形，且必须与 xy 轴平行，问一颗炸弹炸毁的最大价值。N<=10000, x, y <=5000

解法

- 按照上述方法预处理二维前缀和，枚举每个正方形区域即可。

例题

题目牛客-校门外的树

- 长度为 L 的马路上有一排树，相邻间隔 1 米，M 个区域要建地铁，用起始点和终止点表示，现在要将这些区域中的树移走，试问还剩多少棵树。L <= 100000, M <= 100000。

例题

题目牛客-校门外的树

- 长度为 L 的马路上有一排树，相邻间隔 1 米，M 个区域要建地铁，用起始点和终止点表示，现在要将这些区域中的树一走，试问还剩多少棵树。L <= 100000, M <= 100000。

解法

- 这个问题的难点在与区间合并时，要考虑覆盖的情况。可以采用线段树或树状数组等数据结构来维护。或者对区间进行排序来统计。
- 差分的思路则更加简单，对于区间 [l,r]，在 a[l] 的位置 ++，在 a[r+1] 的位置−，统计 a[i] 的前缀和 sum，那么 sum 为 0 的位置就是没被区间覆盖即有树的位置。

考虑 $L <= 10^9$ 时怎么做？

目录

分而治之——分治算法

Description

分治可以认为是一种将问题规模等分成 k 个子问题，然后解决子问题，最终合并子问题的算法。

Character

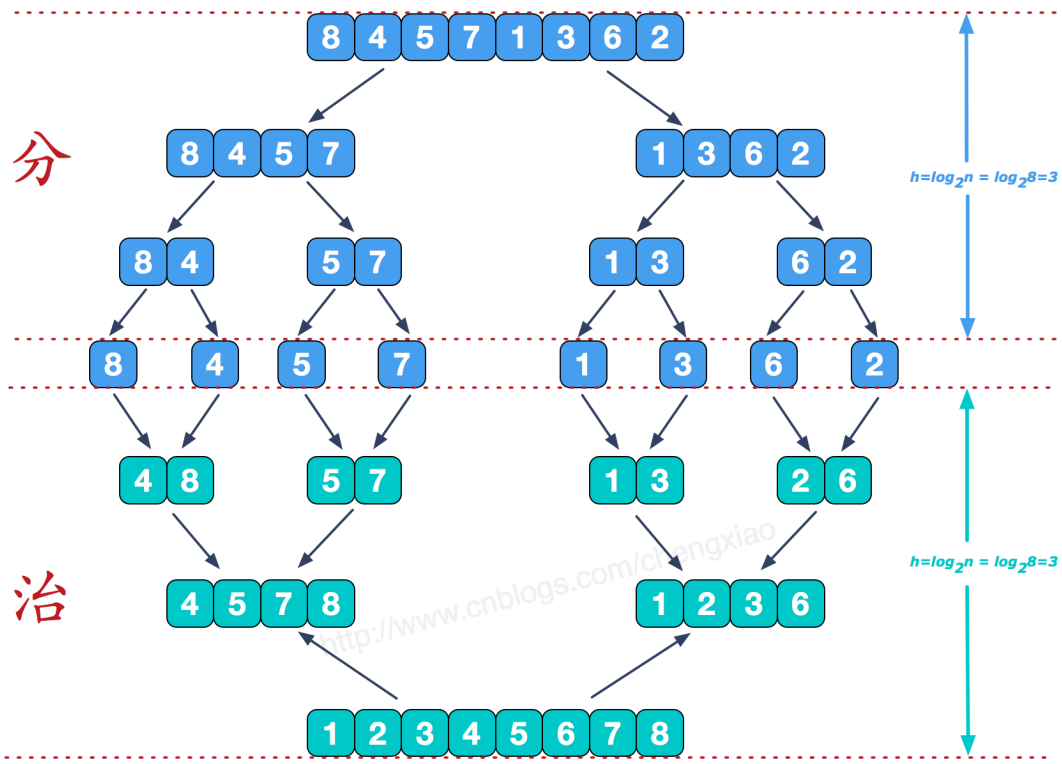
1. 分治算法需要在分割问题和合并问题的时候，不需要花费分多时间。
2. 分割问题的时间是 $O(1)$ 的，算法进行 $\log n$ 次分割，最后的复杂度为 $O(logn)$，如二分检索
3. 分割问题的复杂度为 $O(n)$，算法的复杂度通常为 $O(nlogn)$，如归并排序，二分答案

目录

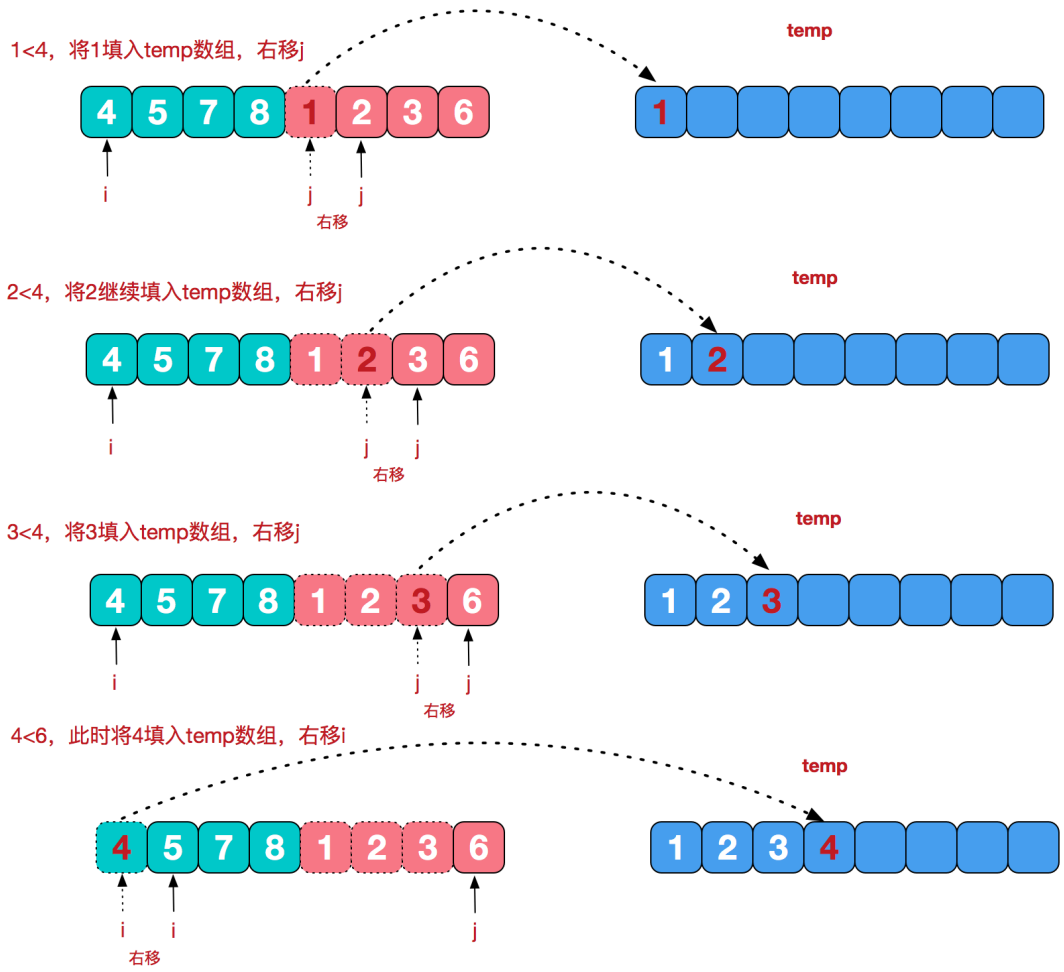
归并排序

Description

归并排序是利用分治思想实现的，将大的排序问题分成给两个小问题排序，两个有序的数组可以在 $O(n)$ 的时间内进行合并，复杂度 $O(nlogn)$

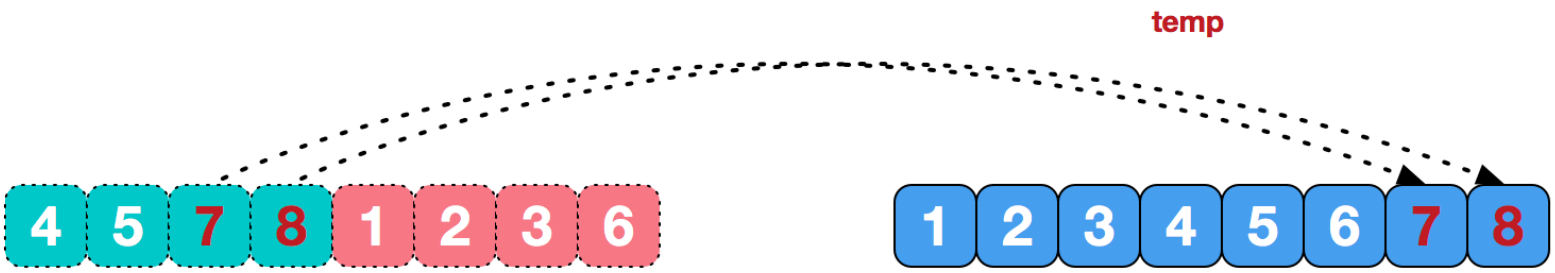


归并排序

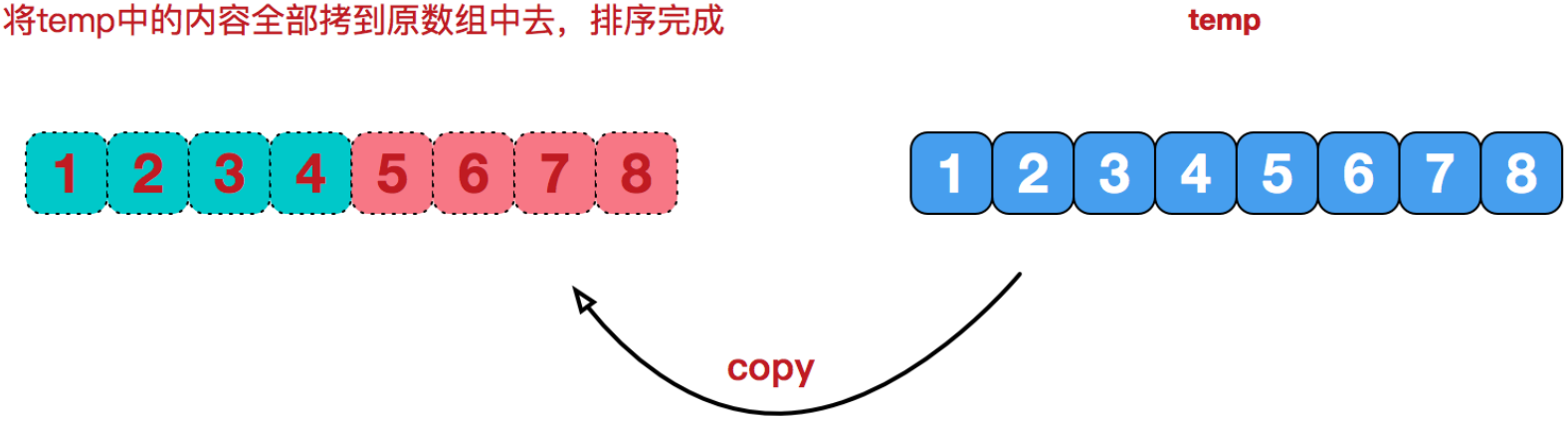

1<4，将1填入temp数组，右移j
temp
4 5 7 8 1 2 3 6
1
i
j
j
右移
2<4，将2继续填入temp数组，右移j
temp
4 5 7 8 1 2 3 6
1 2
i
j
j
右移
3<4，将3填入temp数组，右移j
temp
4 5 7 8 1 2 3 6
1 2 3
i
j
j
右移
4<6，此时将4填入temp数组，右移i
temp
4 5 7 8 1 2 3 6
1 2 3 4
i
i
右移
j

归并排序

继续重复这种比较+填入的步骤，直到右子序列已经填完，这时将左边剩余的7和8依次填入



最后，将temp中的内容全部拷到原数组中去，排序完成



归并排序

```
1   void mergesort(int l,int r,int *a){
2       if(l==r) return;
3       int mid=(l+r)/2;
4       mergesort(l,mid);
5       mergesort(mid+1,r);
6       int l1=l,l2=mid+1;
7       int l3=l;
8       while(l1<=mid&&l2<=r){
9           if(a[l1]<a[l2]) tmp[l3++]=a[l1];
10          else tmp[l3++]=a[l2];
11      }
12      while(l1<=mid) tmp[l3++]=a[l1];
13      while(l2<=mid) tmp[l3++]=a[l2];
14      for(int i=l;i<=r;i++) a[i]=tmp[i];
15  }
```

归并排序

luogu P1309

https://www.luogu.com.cn/problem/P1309

Solve

直接排序即可，但是要用快速高效的排序方法，本题以 sort 和归并为例。

目录

二分答案

Description

二分答案实际上也是二分检索的一种，主要利用单调性。我们通过下面这个例子来学习。

Problem

将 n 个数分成连续的 k 段，使得每段和的最大值尽可能的小

Solve

1. 我们来想这样的性质，如果每一段的和都比较大，那么段数就会少，如果每一段的和小，那么段数就会多，这是一个反比例的单调函数。
2. 所以我们就可以将枚举的过程转化为二分检索答案，然后扫描一遍数组进行判断当前解是否可行，复杂度 $O(nlogn)$。

目录

目录

离散化

回顾

回顾刚刚有道例题《校门外的树》，我们提出一个疑问当 $L <= 10^9$ 时怎么做？

L 很大面临的问题是，没办法枚举 L 也没办法开 L 维数组。
但是我们注意到，M 的值并没有变化，有意义的区间个数仍为 10^5 个，因此，我们是否可以将枚举 L 转化成枚举 M？

离散化

回顾

回顾刚刚有道例题《校门外的树》，我们提出一个疑问当 $L <= 10^9$ 时怎么做？

- 离散化是将离散的散点映射成相对连续的值，将数据范围缩小并保留其顺序性质。
- 例如数组 $[100, 200, 300, 500, 400]$ 完全可以映射到 $[1, 2, 3, 5, 4]$，即可保留其大小顺序意义，当需要计算时只需要映射回去即可。
- 实现方法是，先对原数组排序，并去重，将对应的排名与数值进行一一映射

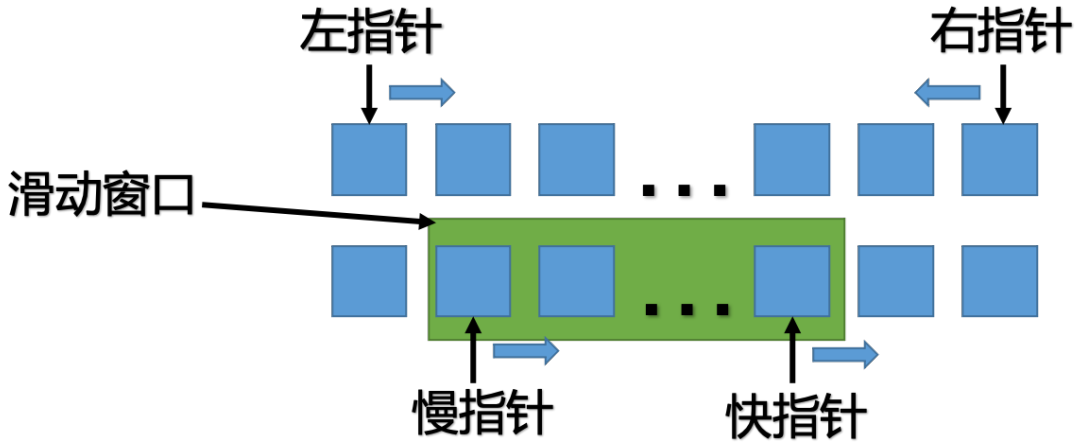
目录

尺取法

基本思想

在有序序列上设置两个游标（指针），一头一尾或一快一慢，进行反向扫描或同向扫描。

- 反向扫描：左指针、右指针同时向中间移动，直到会和
- 同向扫描：快指针和慢指针同时向一个方向移动，形成“滑动窗口”



例题

题目

- 有一个圆形的魔法阵，圆形法阵的圆周上有 N 个点。当法阵圆周上的某四个不同的点恰好能组成矩形时，法阵的能产生 1 单位的能量。（注：若至少存在一个节点不相同，则认为两个矩形不相同，同一个点可以被统计到多个矩形中）
- 现在，已知相邻两点间的弧长均为正整数，按照以某一点为起点顺序给出相邻两点间弧长，求这个法阵能产生多少单位的能量。

解法

简单解法

- 在求完前缀和后，显然题目要求的就是有多少个区间，区间和为总和的一半
- 暴力的枚举区间左右端点的复杂度 $O(n^2)$

双指针解法

- 展环为链，维护两个指针 left、right，限制指针内区间和不超过 sum/2。当区间和小时，首先移动 right 指针至区间和大于等于 sum/2，然后移动 left 指针，缩小区间和。这样当两个指针遍历完一遍后，一定能截取所有满足条件的区间。
- 时间复杂度怎么计算呢？因为每个指针都只遍历每个位置一次，所以复杂度至多是 2*n，即总复杂度 $O(2 * n)$。

目录

Thank you! Any Question?