

- 给一个有序数组，求中位数
- 随时找到数据流的中位数
- 给一堆ip区间和每个区间对应的地点（区间不重合），再任意给一个具体的ip，查出对应的地点。
- 上题中，若区间重合，怎么做。
- 快排原理
- 三角形，圆内等概率随机生成一个点
- DFS
- 并查集
- 线段树
- 一个数组，其中一个数字出现的次数大于1/2，找出这个数字
- N个数字，返回出现频率前K高的数字
- 一个平面凸多边形，一个点在不在多边形内
- 最小路径算法（Dijkstra和floyd算法）
- 无序数组n中大小前K个数据
- 给你两个字符串，比如abc和aeabbqqqcc，问第二个串中包含多少个第一个串，可以不连续，并且任意两个串不能有公共部分，例子答案是2个
- 给你一个无序数组，找两个元素ai和aj，使得ai-aj最大，并且i>j
- 给你一个数组，每个数组有一个正数ai，表示从i位置可以一步往后走到i+ai，当然也可以一步从i走到i+1，问从1走到n最少几步
- 模型太多了没法全加载到内存，当需要一个模型但是没在内存中时就得去加载它，并且释放掉一些已经加载的模型，问怎么优化这个过程
- 怎么均匀洗一副扑克牌
- 排瓷砖问题
- 怎么快速获得一个队列中的最大元素
- 链表排序
- B+树如何保持树平衡的
- 一个二维图，如何优化一个大圆包含尽量多的小圆
- 多个有序链表合并
- 红黑树是什么结构，和二叉平衡树的区别呢，map为什么用红黑树而不用二叉平衡树。
- 一致性哈希。
- 智能指针实现
- 设想产生很多要求保序的请求从多个机器上发到一个多线程的代理上，再由代理调用分布式的数据库，怎么保证这个过程顺序不乱。
- 求一个数组左边之和最接近右边之和的节点。
- 反转链表
- 判断链表是否有环，求环入口节点
- 图的遍历

- 50个红球，50个白球，分别放到两个容器里面，随机选择一个容器，从中取出一个球，如何放球可以使得拿出红球的概率最高
- LRU
- 现在有一个Timer定时器，函数可以注册一个延时，Timer记录下来，从注册开始经过指定延时之后执行这个函数每个一段时间Timer刷新，判断哪个函数到时间了就执行，记录应该用什么数据结构保存？
- 二叉搜索树的实现
- 递归的本质？
- 给一个无限长链表，怎么随机抽K个数，保证每个数抽到概率一致
- 给你一个数，字符串的，返回浮点数（先判合法性、再转
- C语言实现函数重载
- 用 C 模拟虚函数
- 吃鸡地图高度转换成 $[0, 1.0]$ 的数组 然后提需求修改
- 平面吃鸡判断击中谁
- 一个vector当做无序容器储存上万数据，删除中间元素，怎么实现 $O(1)$ 复杂度
- 判断一棵树是否是对称的
- mysql 怎么增加一个索引？
- 中序遍历下一个节点怎么计算
- 一个链表，只知道指向这个节点的指针，怎么删除这个节点（将下一个节点的值复制到当前节点，并删除下一个节点）
- 抢劫犯算法（leetcode 198-House Robber（抢劫犯））
- 空间有三个方块，怎么用一个平面将他们切分成大小相同的两块（三个方块中心点连接成一个面）
- 游戏排行榜怎么实现
- 1) 单字符串压缩：
- 2) 多字符串压缩
- 字符串求出最长不重复子串
- x轴上有n个点，已知每个点的位置p和速度v（正表示向右，负表示向左），每当两个点相碰就消失，问最后碰撞的时间t和两个点
- 一个地图中有许多玩家，如何设计算法，最有效的输出所有玩家周围的一部分玩家的信息。(需要考虑玩家运动的情况)
- $n*n$ 矩阵，从左上角走到右下角（右移、下移）
- 长度为n数组中随机取m个数据，不可重复取，时间复杂度？
- 两个矩形怎么判断重叠
- 字符串解压压缩
- 迷宫动态规划
- 字符串转换最小代价
- 最近点对问题
- 拓扑排序
- 有一个瓶子里面的药丸质量多一些，称一次找出那个瓶子，智力题。

- 一千万游戏玩家，有一个积分排行榜，积分范围在0-999999，找到重复的积分；考虑内存不足，空间复杂度尽量低，如果内存充足
- 环形加油站问题
- 设计一个定时器
- 证明洗牌的随机性

给一个有序数组，求中位数

定义中位数为一个有序数组 $(\text{len}(\text{array})+1)/2$ 处的元素

1. 排序取中值
2. 利用快排的思想
 - 1) 先进行一趟快排，使得 div 左边的值都比 $\text{arr}[\text{div}]$ 小， div 右边的值都比 $\text{arr}[\text{div}]$ 大，但是这个 div 的位置是不确定的，可能位于中间，可能偏左或者偏右；
 - 2) 计算出 mid 所在的下标，如果是奇数则 $\text{mid} = (\text{size} + 1) / 2$ ，如果是偶数则 $\text{mid} = \text{size} / 2$ ；
 - 3) 此时需要比较 mid 和 div 所在的位置。如果 mid 在 div 所在位置的左边，此时就要递归去左半区间查找；如果 mid 在 div 的右边，此时就要递归去右半区间查找；如果恰好相等则说明 div/mid 所在的位置就是中位数。

```

int PartSort(int *arr, int start, int end)
{
    int left = start;
    int right = end;
    int key = arr[end];    //选取关键字
    while (left < right)
    {
        while (left < right && arr[left] <= key)    //左边找比key大的值
        {
            ++left;
        }
        while (left < right && arr[right] >= key)    //右边找比key小的值
        {
            --right;
        }
        if (left < right)
        {
            swap(arr[left], arr[right]);    //找到之后交换左右的值
        }
    }
    swap(arr[right], arr[end]);
    return left;
}

//求一个无序数组的中位数
int GetMidNumNoSort1(int *arr, int size)
{
    assert(arr);
    int start = 0;
    int end = size - 1;
    int mid = (size - 1) / 2;
    int div = PartSort(arr, start, end);
    while (div != mid)
    {
        if (mid < div)    //左半区间找
            div = PartSort(arr, start, div - 1);
        else    //右半区间找
            div = PartSort(arr, div + 1, end);
    }
    return arr[mid];    //找到了
}

```

3. 建堆的思想

步骤 1：如果数组长度为奇数则取数组的前 $(\text{len}(\text{array})+1)/2$ 个元素建立一个最小堆，如果为偶数则取 $(\text{len}(\text{array})+1)/2 + 1$ 个元素建立一个最小堆。

步骤 2：遍历剩余元素，如果该元素小于堆顶元素，则丢弃或不作处理；如果该元素大于堆顶元素，则将其取代堆顶元素，并且重新调整当前堆为最小堆。

步骤 3：遍历结束后，返回堆顶元素，它就是所要寻找的中位数。

随时找到数据流的中位数

用一个大根堆和一个小根堆

大根堆中含有接收的所有数中较小的一半，并且按大根堆的方式组织起来，那么大根堆的堆顶就是较小一半的数中最大的一个。

小根堆中含有接收的所有数中较大的一半，并且按小根堆的方式组织起来，那么小根堆的堆顶就是较大一半的数中最小的一个。

中位数就是两个堆顶相加除以2。

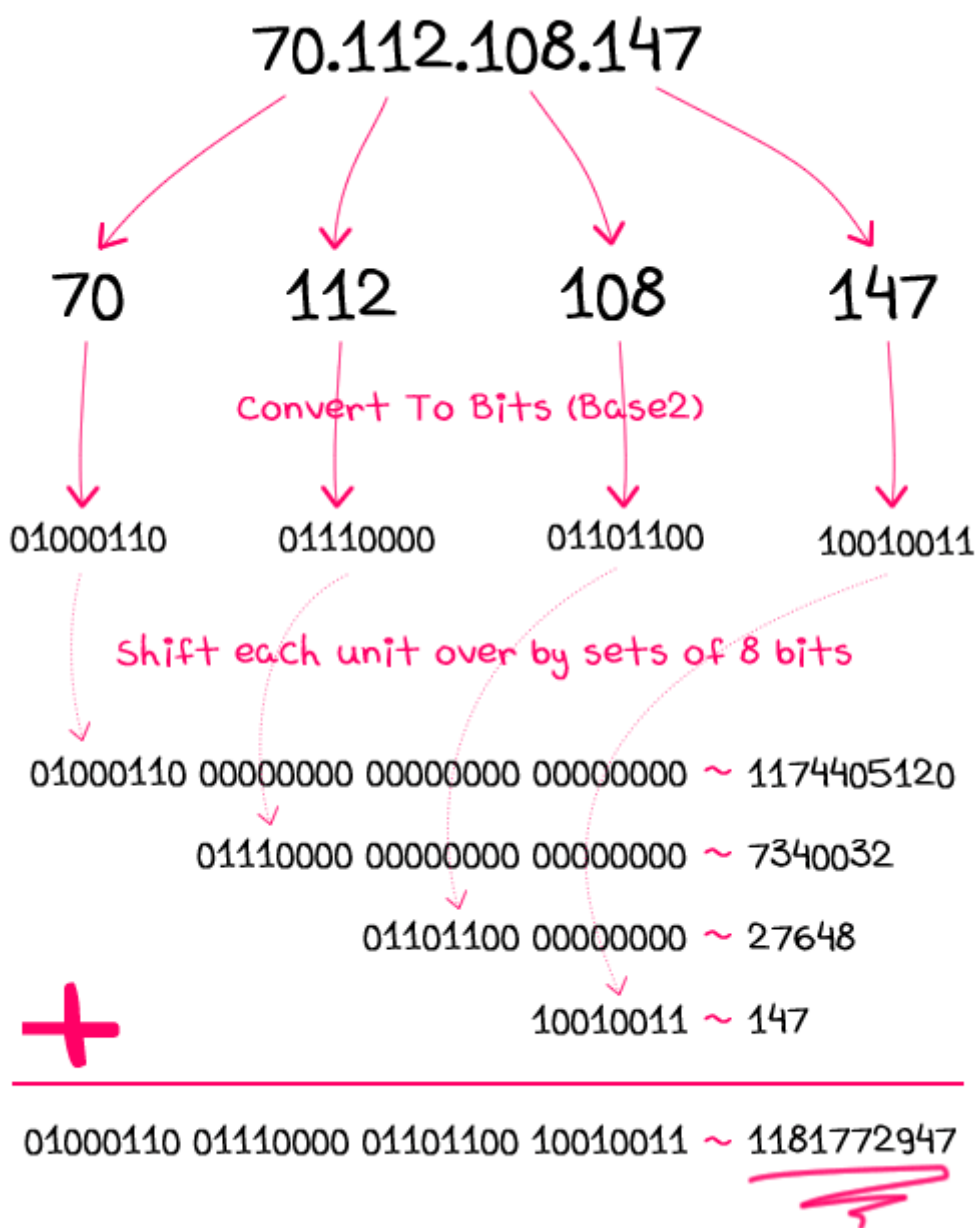
要满足：

1. 如果大根堆的size比小根堆的size大2，那么从大根堆里将堆顶弹出并放入小根堆里；
2. 如果小根堆的size比大根堆的size大2，那么从小根堆里将堆顶弹出并放入大根堆里。

给一堆**ip**区间和每个区间对应的地点（区间不重合），再任意给一个具体的**ip**，查出对应的地点。

使用二分查找的变种

IP地址转化为32位整型数，将起始地址从小到大进行排序。



要查找某个IP对应的地区时，通过二分查找，找到最后一个起始地址IP小于等于这个IP的IP区间。
若IP在这个区间内，则找到；否则就是没有找到

二分查找的变种问题：

https://blog.csdn.net/qq_25800311/article/details/89328326

上题中，若区间重合，怎么做。

总体步骤和上面一样。

最后在验证的时候，需要判断得到结果的重合的区间。

快排原理

三角形，圆内等概率随机生成一个点

DFS

并查集

线段树

一个数组，其中一个数字出现的次数大于 $1/2$ ，找出这个数字

N个数字，返回出现频率前**K**高的数字

一个平面凸多边形，一个点在不在多边形内

最小路径算法（**Dijkstra**和**floyd**算法）

无序数组**n**中大小前**K**个数据

快排的思想：时间复杂度

堆：时间复杂度

内存很小怎么办

给你两个字符串，比如**abc**和**aeabbqqqcc**，问第二个串中包含多少个第一个串，可以不连续，并且任意两个串不能有公共部分，例子答案是**2**个

这个题和PAT上的一个题差不多， $O(n)$ 扫一遍就出答案了

给你一个无序数组，找两个元素 **a_i** 和 **a_j** ，使得 **$a_i - a_j$** 最大，并且 **$i > j$**

直接贪心就行了吧，但是面试官要求用dp搞。。把序列**a**当成一个前缀和，还原出原始序列，求最大连续子段和

给你一个数组，每个数组有一个正数 **a_i** ，表示从**i**位置可以一步往后走到 **$i + a_i$** ，当然也可以一步从**i**走到 **$i + 1$** ，问从**1**走到**n**最少几步

比较简单的dp吧，或者直接BFS一下

模型太多了没法全加载到内存，当需要一个模型但是没在内存中时就得去加载它，并且释放掉一些已经加载的模型，问怎么优化这个过程

就是LRU、LFU那些策略吧

怎么均匀洗一副扑克牌

排瓷砖问题

怎么快速获得一个队列中的最大元素

滑动窗口的最值问题

链表排序

B+树如何保持树平衡的

一个二维图，如何优化一个大圆包含尽量多的小圆

多个有序链表合并

红黑树是什么结构，和二叉平衡树的区别呢，**map**为什么用红黑树而不用二叉平衡树。

一致性哈希。

智能指针实现

设想产生很多要求保序的请求从多个机器上发到一个多线程的代理上，再由代理调用分布式的数据库，怎么保证这个过程中的顺序不乱。

求一个数组左边之和最接近右边之和的节点。

反转链表

判断链表是否有环，求环入口节点

图的遍历

50个红球，50个白球，分别放到两个容器里面，随机选择一个容器，从中取出一个球，如何放球可以使得拿出红球的概率最高

一个箱子放1个红球 另一个放49红球和50篮球 拿到红球概率= $0.5+49/99$ 约等于0.75

LRU

现在有一个**Timer**定时器，函数可以注册一个延时，**Timer**记录下来，从注册开始经过指定延时之后执行这个函数每个一段时间**Timer**刷新，判断哪个函数到时间了就执行，记录应该用什么数据结构保存？

用堆就可以了

二叉搜索树的实现

递归的本质？

给一个无限长链表，怎么随机抽**K**个数，保证每个数抽到概率一致

给你一个数，字符串的，返回浮点数（先判合法性、再转

合法性解释

如果是16进制的科学计数法呢？

C语言实现函数重载

c语言中不允许有同名函数，因为编译时函数命名是一样的，不像c++会添加参数类型和返回类型作为函数编译后的名称，进而实现重载。如果要用c语言显现函数重载，可通过以下方式来实现：

- 使用函数指针来实现，重载的函数不能使用同名称，只是类似的实现了函数重载功能
- 重载函数使用可变参数，方式如打开文件open函数
- gcc有内置函数，程序使用编译函数可以实现函数重载

```

#include<stdio.h>
void func_int(void * a)
{
    printf("%d\n",*(int*)a); //输出int类型，注意 void * 转化为int
}
void func_double(void * b)
{
    printf("%.2f\n",*(double*)b);
}
typedef void (*ptr)(void *); //typedef申明一个函数指针
void c_func(ptr p,void *param)
{
    p(param);                //调用对应函数
}
int main()
{
    int a = 23;
    double b = 23.23;
    c_func(func_int,&a);
    c_func(func_double,&b);
    return 0;
}

```

用 C 模拟虚函数

吃鸡地图高度转换成 **[0, 1.0]**的数组 然后提需求修改

平面吃鸡判断击中谁

一个**vector**当做无序容器储存上万数据，删除中间元素，怎么实现**O(1)**复杂度

直接swap pop_back

判断一棵树是否是对称的

mysql 怎么增加一个索引？

中序遍历下一个节点怎么计算

一个链表，只知道指向这个节点的指针，怎么删除这个节点（将下一个节点的值复制到当前节点，并删除下一个节点）

抢劫犯算法（leetcode 198-House Robber（抢劫犯））

空间有三个方块，怎么用一个平面将他们切分成大小相同的两块（三个方块中心点连接成一个面）

游戏排行榜怎么实现

1) 单字符串压缩：

输入：ABBBCCD，输出AB3C2D

2) 多字符串压缩

输入：AABCABCD，输出A(ABC)2D

字符串求出最长不重复子串

x轴上有**n**个点，已知每个点的位置**p**和速度**v**（正表示向右，负表示向左），每当两个点相碰就消失，问最后碰撞的时间**t**和两个点

一个地图中有许多玩家，如何设计算法，最有效的输出所有玩家周围的一部分玩家的信息。(需要考虑玩家运动的情况)

A：暴力计算所有玩家距离太耗时，提示没听清：拓扑排序还是topk？

n*n矩阵，从左上角走到右下角（右移、下移）

长度为**n**数组中随机取**m**个数据，不可重复取，时间复杂度？

A：用bool或set记录已取下标

两个矩形怎么判断重叠

字符串解压压缩

迷宫动态规划

字符串转换最小代价

最近点对问题

拓扑排序

有一个瓶子里面的药丸质量多一些，称一次找出那个瓶子，智力题。

一千万游戏玩家，有一个积分排行榜，积分范围在**0-99999**，找到重复的积分；考虑内存不足，空间复杂度尽量低，如果内存充足

环形加油站问题

设计一个定时器

证明洗牌的随机性