1. 说出常见数据集合中三种典型类型的数据值

String Integer Long

1. 说出常见数据集合中的三种典型结构。

List Map Set Queue Tree

1. 说出常见数据集合中的三种典型操作。

add(E e)

remove(Object o)

contains(Object o)

isEmpty()

1. 我们为什么要学习数据结构？

We learn data structures to efficiently organize, manage, and store data for optimized access and modification in software applications.

1. Can we program data structure in Java/C?

Yes.

1. 为什么我们坚持算法必须终止？

We insist an algorithm must terminate to ensure it produces a result in a finite amount of time, making it useful and practical.

1. 为什么我们坚持算法必须精确？

Ensure it produces correct and predictable results.

1. 为什么算法中的指令是按顺序编写的？

Instructions in an algorithm are written in a sequence to provide a clear, step-by-step procedure for solving a problem or performing a task.

1. 空间效率和时间效率都是用于评估算法（和数据结构）性能的指标。 T
2. 动态数据结构通常更节省空间。 F
3. 静态数据结构通常更具时间效率。 F
4. 信息隐藏是软件组件的用户只需要知道如何初始化和访问组件的基本细节，而不需要知道实现的细节 T
5. 说出3种可以在线性集合Java编程下实现的集合类型

List Queue Deque

1. 说出3个可以在Java线性集合编程下实现的操作

add(E e)

remove(Object o)

contains(Object o)

1. 命名2种可以在分层集合Java编程下实现的集合类型

TreeSet TreeMap

1. 说出4个可以在分层集合Java编程下实现的操作

Add() put() remove(Object o/key)

1. 什么是库函数

软件“库”是预先编写的代码、函数、类和例程的集合，开发人员可以使用它们来执行常见任务、实现功能和解决特定问题，而无需从头开始编写代码。库被设计为可重用和模块化，允许开发人员将它们集成到自己的程序中，以增强功能并提高效率。

1. 什么是Java中的包

在Java中，“包”是组织一组相关类和接口的命名空间，包有助于避免命名冲突，管理权限改造，更容易以结构化的方式定位和使用类和接口，它们还通过将相关类和接口分组在一起来促进模块化编程。

1. 说出Java的IO库

Java.io.\*

1. 说出java的GUI库

Java.awt javafx java.swing

1. Java怎么导包 import
2. ArrayList中存储什么？

In Java, an ArrayList stores objects. It's a dynamic array implementation, meaning it can dynamically resize itself to accommodate the addition or removal of elements.

1. ArrayList怎么跟踪size

using an integer variable called size. This size variable represents the number of elements currently stored in the ArrayList.

1. ArrayList必要时怎么扩容？

it dynamically increases its capacity by reallocating its internal array. This process is known as "resizing" or "growing" the ArrayList.

1. Java指定了队列接口。（T或F） T
2. Java对“优先级队列”没有任何类支持 F
3. Queue接口下的peek() 如果队列为空会抛出异常 F
4. 如果队列为空，Queue接口下的poll() 抛出异常 F
5. Queue接口下有一个element() 方法 T
6. Iterable是配备Iterator的类的接口规范。 T
7. 迭代器是可以生成迭代元素的类的接口规范。T
8. comparable类下定义的对象将具有“自然排序” T
9. 可以使用哪种方法比较在comparable类下声明的对象？ compareTo()
10. compareTo方法的签名是什么？

public int compareTo(T o);

1. 哪种方法可用于对comparable对象列表进行排序？ Collections.sort()
2. Comparator是一个可以比较其他对象的对象。 T
3. What is the signature for the compare() method?

int compare(T o1, T o2);

1. 一个可比类可以实现多个比较器。 T
2. 当出现问题时，一种方法会做什么？

抛出异常

1. 当出现问题时，异常提供本地退出。 T
2. 说出三种可以抛异常的情况

ArithmeticException

NullPointerException

ArrayIndexOutOfBoundsException

1. 怎么捕获异常？

Try-catch / Throws / throw

1. 异常有哪些类型

Checked Exceptions，运行时异常，Error

1. 异常处理程序可以使用异常对象中的信息吗？

可以，可以直接logger

1. “e. getMessage（）”在e是异常对象的情况下做什么？

获取异常的详细信息，返回一个String

1. 不需要处理RuntimeExceptions。 T
2. IOException不必处理。 F
3. 当我们尝试将元素添加到不可变列表时会发生什么？ UnsupportedOperationException
4. Java接口是否提供构造函数？ 没有
5. ArrayList实现了哪个接口？

List Serializable Collection Iterable

1. List<E>集成了哪些Java接口？Collection<E>
2. 抽象类的主要特征是什么？抽象
3. Can an abstract class be instantiated? 不可以哦
4. 可以在类中定义抽象方法以节省实现工作。T
5. 当我们从ArrayList中删除元素时，实现的关键问题是什么？

并发，线程不安全。索引要正确，还需要维护size，并且由于是动态数组，删除中间元素需要让其余元素移动。

1. 当我们从ArrayList中添加元素时，实现的关键问题是什么？ 同上
2. Remove()在迭代器实现中是强制性的。F
3. ArrayList如何在其实现中使用“类型参数”？ <>
4. ArrayList.remove()将删除哪个元素？

只能删除指定索引或对象的元素

1. ArrayList.next（）在返回列表中的下一个元素之前检查什么？ 索引有无越界
2. ArrayList.remove() 如何确保每次调用next() 后只能删除1个元素？

Iterator接口的remove方法保证每次调用next

时只调用一次。

1. 如果2个或多个迭代器在同一个ArrayList下同时运行会发生什么？

ConcurrentModificationException

1. O(log(n)) < O(sqrt(n)) T
2. O(n^n) < O(n!) F
3. O(2^n) < O(n^n) T
4. 在分析算法的成本时，循环通常是重点。 T
5. 以下哪项操作更昂贵？-从文件中读取一行-从用户中读取一行

磁盘IO更昂贵，键盘通常是采用中断操作来实现IO的。

1. 最坏情况下的成本分析通常比平均成本分析更困难。 T
2. 设计递归函数的关键步骤是什么？

递归停止条件，分治，划分成重复的小问题

1. 每个递归函数都可以重写为迭代函数。 T
2. 你什么时候为“黑盒”测试编写测试用例？在实现之前还是之后？

实现之前

1. 解释为什么队列的数组实现很慢。

Fixed Size and Resizing

Element Shifting

In a straightforward array implementation, when you dequeue an element, you might need to shift all remaining elements one position to the left to fill the gap at the front. This shifting operation takes O(n) time in the worst case, making dequeue operations inefficient.

1. LinkedList允许数据通过什么删除？

改变指针的指向

1. 定义引用/指针。

指向下一个对象的地址

1. 内存管理中垃圾回收机制的目的是什么？

回收不需要的资源，节省内存

1. 树表示高效的一维数据结构 F
2. 树中的叶节点没有子节点。 T
3. 二叉树对其兄弟节点没有排序。 F
4. Name 3 applications for tree.

File Systems

Database Indexing

Compiler Design and Syntax Trees

1. 递归调用之间的关系可以用什么类型的树来表示？

递归树（Recursion Tree）