

给一个 MergedSortedStream 的 class 包装一定数量的 SortedStream, Stream 都有 hasNext() 和 next() 的方法, SortedStream 定义是从这个流读取的数据一定是升序的, 现在让我实现 MergedSortedStream 的 hasNext() 和 next() 方法, 从而达到 MergedSortedStream 的 next() 返回的数据也是升序的。大概思路跟 leetcode 的 merge k list 一样, 用 min heap 来做, 但是这里有个问题就是 SortedStream 缺一个 peek() 方法, 我一开始没往定义一个 wrapper class 那方面想, 被面试官提醒才突然想起来以前在地里看过的 GOOGLE 面经有考过 PeekIterator 的实现, 如出一辙。慌慌忙忙写完了 wrapper class 时间已经到了

1) Why do we have interfaces?

2) What does synchronized do?

3)

```
interface Iterator {
```

```
    Object next();
```

```
    boolean hasNext();
```

```
}
```

```
[3, 3, 2, 4, 5]
```

```
while (hasNext()){
```

```
    next();
```

```
}
```

```
interface Stream {
```

```
    Object next();
```

```
    boolean hasNext();
```

```
}
```

```
[3, 3, 2, 4, ... ]
```

```
/** Precondition: ever call to .next() returns the minimal value in the stream */
```

```
interface SortedStream {
```

```
    Comparables next();
```

```
    boolean hasNext();
```

```
}
```

```
[2, 3, 3, 4, ... ]
```

```
while (hasNext()){
```

```
    next();
```

```
}
```

```
List<SortedStream> list =
```

```
[new SortedStream([2, 3, 3, 4, ... ])
```

```
[2, ... ]
```

如何实现 iterator 的 peek。

iterator 实现 next hasNext, flatten list 交替输出 [[7 8 9] [] [2 3 5] [1 19]] ---> 7 2 1 8 3 19  
9 5

OO 题, java 中的 iterator 使用的是什么 design pattern? decorator 就是个 inner class。写出代码并且使得 iterator 可以查看前一个遍历的内容。有点记不清了,大致就是包装一下 iterator 多出一个存储单元记录内容。

写一个 PeekIterator, 包装一个普通的 Iterator, 要实现 peek()方法, 返回当前 iterator 指向的元素, 但是不能移动它。除此之外也要实现 hasNext()和 next()方法。第二题, 给一个 BST 的 Node, 有父节点的指针, 求该 Node 的下一个 Node。这一面感觉不错。

写一个 EvenIterator(汗, 这年头这么流行考 iterator 么? ), 包装一个普通的 Iterator, 只返回 even number。要求实现 next()和 hasNext()方法。这一面感觉不太好, 没想到先写个简单的方法。最后时间到了还是有 bug 没有修改完。

写 jump iterator 类, 构造函数传入一个普通的 iterator, 然后实现 next(), hasNext(). next()返回传入 iterator 的 next().next(), 就是每次跳过一个元素输出。

然后再实现一个 rotatetiterator(), 构造函数传入 List<Iterator<T>>, 实现 next(), hasNext(). 例如:

传入的三个 iterator 里面的值分别是[[1,2,3],[4,5,6], [7,8]], 那 rotatetiterator 的 next()应该输出 [1,4,7,2,5,8,3,6]. 就是竖着遍历每个 iterator 输出, 如果当前的 iterator 没有了, 就跳到下一个。

扩展的斐波那契数列。  $S_n = S_{n-j} + S_{n-k}$  ( $0 < j < k$ ). 写一个 generator 类。有实现 next()方法。

写一个 Stream 的 interface, 就是有 generic, 有 peek(), next(), hasNext(), append()方法. 然后写一个 merge List of sorted stream, 就是的 k-way merge. 然后因为是 generic, 传入参数要包括一个 comparator.

写个 stream 的 merge 的代码。