

Java - 内存资源分配 - 有一个简易内存池

题目描述:

有一个简易内存池，内存按照大小粒度分类，每个粒度有若干个可用内存资源，用户会进行一系列内存申请，需要按需分配内存池中的资源，返回申请结果成功失败列表。分配规则如下：

- 1、分配的内存要大于等于内存申请量，存在满足需求的内存就必须分配，优先分配粒度小的，但内存不能拆分使用。
- 2、需要按申请顺序分配，先申请的先分配。
- 3、有可用内存分配则申请结果为 true，没有可用内存分配则返回 false。

注：不考虑内存释放。

输入描述:

输入为两行字符串：

第一行为内存池资源列表，包含内存粒度数据信息，粒度数据间用逗号分割，一个粒度信息内部用冒号分割，冒号前为内存粒度大小，冒号后为数量。资源列表不大于 1024，每个粒度的数量不大于 4096

第二行为申请列表，申请的内存大小间用逗号分隔。申请列表不大于 100000

如：

64:2,128:1,32:4,1:128

50,36,64,128,127

输出描述:

输出为内存池分配结果。

如：

true,true,true,false,false

补充说明:

示例1

输入: 64:2,128:1,32:4,1:128

50,36,64,128,127

输出: true,true,true,false,false

说明: 内存池资源包含: 64K共2个、128K共1个、32K共4个、1K共128个的内存资源;  
针对50,36,64,128,127的内存申请序列,分配的内存依次是: 64,64,128,NULL,NULL,第三次申请内存时已经将128分配出去,  
因此输出结果是: true,true,true,false,false

```
import java.util.*;
```

```
import java.util.stream.Collectors;
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        String memoryString = sc.nextLine();
```

```
        ArrayList<MemoryInfo> memoryInfoList = new ArrayList<>();
```

```

        if (memoryString != null && !memoryString.equals("")) {
            String[] memoryInfoArray = memoryString.split(",");

            for (String memoryInfoString : memoryInfoArray) {
                String[] infoArray = memoryInfoString.split(":");
                MemoryInfo memoryInfo = new MemoryInfo(Integer.parseInt(infoArray[0]),
                    Integer.parseInt(infoArray[1]));
                memoryInfoList.add(memoryInfo);
            }
        }
        memoryInfoList.sort(Comparator.comparingInt(o -> o.size));
        String applyString = sc.nextLine();
        if (applyString == null || applyString.equals("")) {
            System.out.println("");
            return;
        }
        List<Integer> applyList = Arrays.stream(applyString.split(","))
            .filter(k -> k != null && !k.equals(""))
            .map(Integer::parseInt).collect(Collectors.toList());
        ArrayList<String> res = new ArrayList<>();
        for (Integer applyMemory : applyList) {
            boolean flag = false;
            for (MemoryInfo memoryInfo : memoryInfoList) {
                if (memoryInfo.size >= applyMemory && memoryInfo.num > 0) {
                    flag = true;
                    memoryInfo.setNum(memoryInfo.num - 1);
                    break;
                }
            }
            res.add(flag ? "true" : "false");
        }
        System.out.println(String.join(", ", res));
    }

```

```

public static class MemoryInfo {
    private int size;
    private int num;

    public MemoryInfo(int size, int num) {
        this.size = size;
        this.num = num;
    }
}

```

```
        public int getSize() {  
            return size;  
        }  
  
        public void setSize(int size) {  
            this.size = size;  
        }  
  
        public int getNum() {  
            return num;  
        }  
  
        public void setNum(int num) {  
            this.num = num;  
        }  
    }  
}
```