

正整数分解-cvte-2018网申

笔记本: algorithms
创建时间: 2018/9/21 16:16
作者: zhuhanqing258@foxmail.com
URL: about:blank

更新时间: 2018/9/21 16:53

题目描述:

给定一个包含若干正整数的集合N, 即N中的每个元素均为大于1的正整数, 且每个元素均不重复, 即彼此不相等。

现给定一个目标正整数A, 要求找出所有满足以下条件的集合:

- 1.该集合中的所有元素均来源于N;
- 2.集合中的元素可以重复;
- 3.集合中所有元素相加的和为A。

思路描述:

1.将从1到A的整数分解为两个值相加的和, 并保存这样的两个值的组合。如 $4=1+3=2+2$, 保存(1,3)和(2,2)到整数4的分解组合中。

2.对从1到A的所有整数的分解组合, 进行进一步处理。按从小到大的顺序, 对每个正整数进行进一步分解, 即将原始分解组合中的两个元素中的每一个进行分解, 并重新排列组合。如 $2=1+1, 3=1+2$, 则3可进一步分解为 $3=1+2=1+(1+1)$, 同理 $4=1+3=1+(1+2)=1+(1+(1+1))=2+2=(1+1)+(1+1)$, 以此类推。

3.在上述分解过程中, 若分解组合中存在不属于N的整数的分解方式, 则遗弃该组合。

4.当分解到小于目标正整数A且属于N的最大值时, 将该元素自身也作为一种组合。

5.对于目标正整数, 找出所有的两值之和的组合(不要求两元素均属于N), 并按上述思路进行分解组合。

6.对步骤5中得到的结果进行排查处理, 即剔除不满足条件的组合, 得到最终结果。

实现代码:

```
组合新解
    m addResult(List<List<Integer>>, List<Integer>):void 添加新解, 避免重复
    m combineNewResult(List<List<Integer>>, List<List<Integer>>, List<List<Integer>>):void
    m fastPrint(List<Integer>):void 快速组合新解
递归分解过程
    m findCollections(int, int, int, List<List<Integer>>, List<Integer>):void
    m handleResult(List<List<Integer>>, Map<Integer, List<List<Integer>>>):void 处理结果
    m ifElementResolvable(Integer, Map<Integer, List<List<Integer>>>):boolean
    m init(List<Integer>, int):void
    m main(String[]):void 主程序
    m maxN(List<Integer>, Integer, Integer):boolean
    m printNumberMap(Map<Integer, List<List<Integer>>>):void
    m printResult(List<List<Integer>>):void
    m removeErrorResults(List<List<Integer>>, Integer):void 剔除错误解
解析整数
    m resolveNumberMap(Map<Integer, List<List<Integer>>>, List<Integer>, Integer):void
    m twoListCompareUtil(List<Integer>, List<Integer>):boolean
    f initNumbers:List<Integer> = Arrays.asList(...) 原始集合N
整数分解情况
    f numberMap:Map<Integer, List<List<Integer>>> = new HashMap<>()
    f result:List<List<Integer>> = new ArrayList<>() 结果集
```



Solution2.java

2018/9/21 16:50, 15.0 KB