

【华为OD机试真题2023 JAVA】AI处理器组合

原创

若博豆

于 2023-01-06 10:36:26 发布

1424 收藏 6

分类专栏：[华为OD机试2023 \(JAVA\)](#) 文章标签：[算法](#) [华为](#) [java](#)

华为OD机试2023 (...

该专栏为热榜专栏榜 第1名

¥59.90

¥99.90

750 订阅

213 篇文章

已订阅

超级会员免费看

华为OD机试真题，2023年度机试题库全覆盖，刷题指南点这里

AI处理器组合

知识点数组

时间限制：1s 空间限制：256MB 限定语言：不限

题目描述：

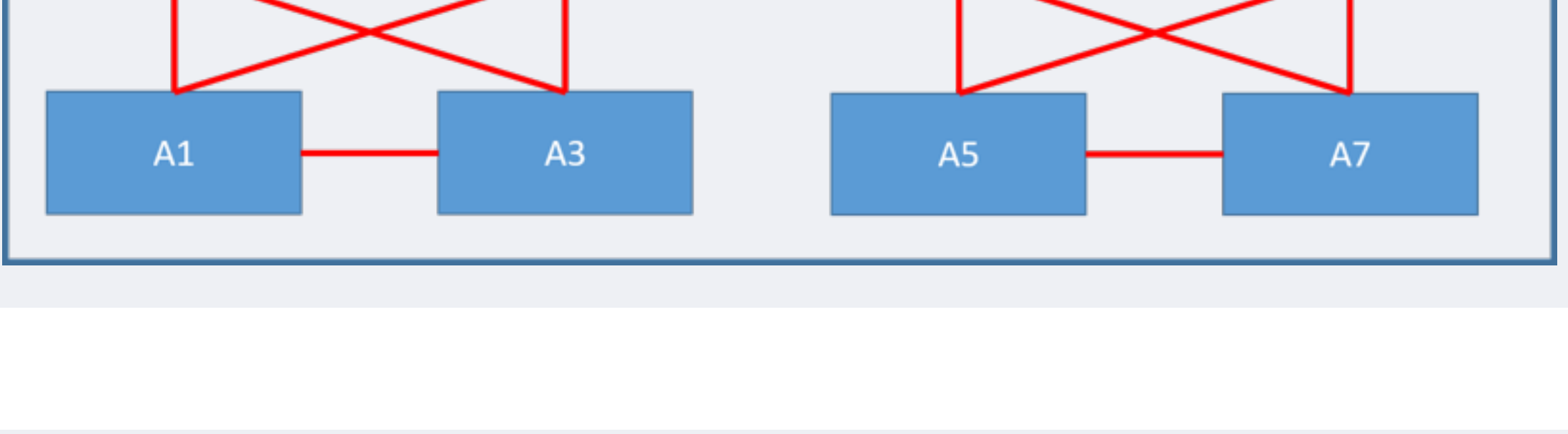
某公司研发了一款高性能AI处理器。每台物理设备具备8颗AI处理器，编号分别为0、1、2、3、4、5、6、7。编号0-3的处理器处于同一个链路中，编号4-7的处理器处于另外一个链路中，不通链路中的处理器不能通信，如下图所示。现给定服务器可用的处理器编号数组array，以及任务申请的处理器数量num，找出符合下列亲和性调度原则的芯片组合。如果不存在符合要求的组合，则返回空列表。

亲和性调度原则：

- 如果申请处理器个数为1，则选择同一链路，剩余可用的处理器数量为1个的最佳，其次是剩余3个的为次佳，然后是剩余2个，最后是剩余4个。
- 如果申请处理器个数为2，则选择同一链路剩余可用的处理器数量2个的为最佳，其次是剩余4个，最后是剩余3个。
- 如果申请处理器个数为4，则必须选择同一链路剩余可用的处理器数量为4个。
- 如果申请处理器个数为8，则申请节点所有8个处理器。

提示：

- 任务申请的处理器数量只能是1、2、4、8
- 编号0-3的处理器处于一个链路，编号4-7的处理器处于另外一个链路。
- 处理器编号唯一，且不存在相同编号处理器



输入描述：

输入包含可用的处理器编号数组array，以及任务申请的处理器数量num两个部分。

第一行为array，第二行为num。例如：

[0, 1, 4, 5, 6, 7]

1

表示当前编号为0、1、4、5、6、7的处理器可用。任务申请1个处理器。

0<= array.length <= 8

0<= array[i] <= 7

num in [1, 2, 4, 8]

输出描述：

输出为组合列表，当array=[0, 1, 4, 5, 6, 7]，num=1时，输出为[[0],[1]]

示例1

输入：

[0, 1, 4, 5, 6, 7]

1

输出：

[[0],[1]]

说明：

根据第一条亲和性调度原则，在剩余两个处理器的链路（0,1,2,3）中选择处理器。由于只有0和1可用，则返回任意一颗处理器即可

示例2

输入：

[0, 1, 4, 5, 6, 7]

4

输出：

[[4, 5, 6, 7]]

说明：

根据第三条亲和性调度原则，必须选择同一链路剩余可用的处理器数量为4个的环。

解题思路：

这题主要还是 [逻辑题](#)。

主要难点是：

需要考虑可用处理器数量的优先级；

需要求出申请处理器的所有可能性。

代码：

```
1 public class Main{
2
3     public static List<Integer> first;    // 第一个链路
4     public static List<Integer> second;    // 第二个链路
5     public static int remainFirst;    // 第一个链路的可用处理器个数
6     public static int remainSecond;    // 第二个链路的可用处理器个数
7     public static List<List<Integer>> tempList = new ArrayList<>();
8
9     public static void main(String[] args) {
10
11         Scanner sc = new Scanner(System.in);
12
13         String[] strings = sc.nextLine().replace("[", "").replace("]", "").split(",");
14
15         int num = sc.nextInt();    // 申请处理器的数量
16
17         first = new ArrayList<>();
18         second = new ArrayList<>();
19         for(String s : strings){
20
21             int i = Integer.valueOf(s.trim());
22             if( i < 4){
23                 first.add(i);
24             }else {
25                 second.add(i);
26             }
27         }
28         remainFirst = first.size();
29         remainSecond = second.size();
30
31         if(num == 1){
32             oneCpu();
33         }else if(num == 2){
34             twoCpu();
35         }if(num == 4){
36             if(remainFirst == 4){
37                 tempList.add(Arrays.asList( 0, 1, 2, 3));
38             }
39             if(remainSecond == 4){
40                 tempList.add(Arrays.asList( 4, 5, 6, 7));
41             }
42         }if(num == 8){
43             if(remainSecond == 4 && remainSecond == 4){
44                 tempList.add(Arrays.asList( 0, 1, 2, 3, 4, 5, 6, 7));
45             }
46         }
47     }
48     System.out.println(tempList.size() == 0 ? "[]" : tempList);
49 }
50
51 /**
52  * 申请处理器个数为1
53  * 则选择同一链路，剩余可用的处理器数量为1个的最佳
54  * 其次是剩余3个的为次佳
55  * 然后是剩余2个
56  * 最后是剩余4个
57 */
58 public static void oneCpu(){
59
60     List<List<Integer>> lists = new ArrayList<>();
61     int[] ints = new int[]{ 1, 3, 2, 4};    // 可用处理器数量的优先级
62     boolean isFit = false;    // 是否有满足的链路
63
64     for(int i : ints){
65         if(i == remainFirst){
66             lists.add(first);
67             isFit = true;
68         }
69         if(i == remainSecond){
70             lists.add(second);
71             isFit = true;
72         }
73         if(isFit){
74             break;
75         }
76     }
77
78     for(int i=0; i<lists.size(); i++){
79         handle( lists.get(i), new ArrayList<>(), 0, 1);
80     }
81 }
82
83 /**
84  * 申请处理器个数为2
85  * 则选择同一链路，剩余可用的处理器数量2个的为最佳
86  * 其次是剩余4个
87  * 最后是剩余3个
88 */
89 public static void twoCpu(){
90
91     List<List<Integer>> lists = new ArrayList<>();
92     int[] ints = new int[]{ 2, 4, 3};    // 可用处理器数量的优先级
93     boolean isFit = false;    // 是否有满足的链路
94
95     for(int i : ints){
96         if(i == remainFirst){
97             lists.add(first);
98             isFit = true;
99         }
100         if(i == remainSecond){
101             lists.add(second);
102             isFit = true;
103         }
104         if(isFit){
105             break;
106         }
107     }
108
109     for(int i=0; i<lists.size(); i++){
110         handle( lists.get(i), new ArrayList<>(), 0, 2);
111     }
112 }
113
114 /**
115  * 申请n个处理器的所有可能性
116  * @param firstOrSecond  连中的链路
117  * @param list            申请的处理器
118  * @param index           处理器的索引
119  * @param n               申请处理器的个数
120 */
121 public static void handle(List<Integer> firstOrSecond, List<Integer> list, int index,
122
123     if(list.size() == n){
124         List<Integer> tempList = new ArrayList<>();
125         tempList.addAll(list);
126         tempList.add(tempList);
127     }else {
128         for( int i=index; i<firstOrSecond.size(); i++){
129
130             list.add(firstOrSecond.get(i));
131             handle(firstOrSecond, list, i +1, n);
132             list.remove(list.size()-1);
133         }
134     }
135 }
136 }
137 }
138 }
139 }
```

📖 文章知识点与官方知识档案匹配，可进一步学习相关知识

算法技能树 > 首页 > 概览 > 40097 人正在系统中学习

华为OD机考100题，真题 06-05

华为OD机考100题

华为OD机试真题.pdf 04-15

华为机试真题（非牛客网试练题）OD考试真题，不定期更新，文档含代码解答

评论 1条 > 写评论

...10北京华为OD机试真题分享——满分_华为od机试题_向着百万年薪努力的... 3-4

字符串的一个子序列是原始字符串删除一些(也可以不删除)字符而不改变剩余字符相对位置形成的新字符串。(例如,"abc"是"aebcyd"的一...

华为OD-机试(三道)_华为od机试题_程乾的博客 3-7

题目三 华为od愉快的周末 小华和小为是很要好的朋友,他们约定周末一起吃饭通过手机交流,他们在地图上选择了多个聚餐地点(由于自然...

华为od前端机试题 (回忆版) LK_315的博客 1566

【华为OD机试真题2023 JAVA】核酸检测人员安排 qq_34465338的博客 1814

核酸检测人员安排 题目描述：在系统、网络均正常的情况下组织核酸采样员和志愿者对人群进行核酸检测筛查，每名采样员的效率不同...

华为OD机考题目_华为od机试题_依城之外的博客 3-9

华为OD机试题目一、入门难度 1.取近似值 描述 写出一个程序,接受一个正浮点数值,输出该数值的近似整数。如果小数点后位数大于...

华为OD机试题_Price Tag238的博客 3-8

在一个团队中输出'we are a team',不在一个团队中输出'we are not a team' 2、c 为其他值,或当前行 a 或 b 的标号小于 1 或者大于 n 时,...

深圳工位出租了，创业者一起办公，沟通起来更方便！ 办公室工位出租 3-12

【2023】华为OD机试真题Java-题目0221-AI处理器组合 最新发布 amos_cloud的博客 347

某公司研发了一款高性能AI处理器。每台物理设备具备8颗AI处理器，编号分别为0、1、2、3、4、5、6、7。编号0-3的处理器处于同一...

华为OD机试 题目一、入门难度 1.取近似值 描述 写出一个程序,接受一个正浮点数值,输出该数值的近似整数。如果小数点后位数大于...

华为OD机试真题 Python 实现【处理器问题/高性能AI处理器】 MISAYAONE的博客 6238

某公司研发了一款高性能AI处理器。每台物理设备具备8颗AI处理器，编号分别为0、1、2、3、4、5、6、7。编号0-3的处理器处于同一...

华为OD机试用Python实现 -【AI处理器组合】(2023-Q1 新题) 梦想橡皮擦，专栏100例写作模式先行者，现象级专... 919

AI处理器组合 题目描述 某公司研发了一款高性能 AI 处理器。 每台物理设备具备 8 颗 AI 处理器， 编号分别为 0、1、2、3、4、5、6、...

[java华为机试] 华为od机试题考试真题_od考试_可乐大数据的博客-CSDN... 3-5

[java华为机试] 华为od机试题考试真题_前言综述_本次od机试考试一共三道题,第一道题和第二道题各100分,第三道题200分。本次考试...

华为OD机试真题大全_用Python 解华为机试题_机试宝典_梦想橡皮擦的... 2-25

2023 年用 Python 语言解华为 OD 机试题_一篇博客找全。 在2023 年,Python 已成为广泛使用的编程语言之一,被广泛应用于数据科学、...

华为OD机试 - 处理器问题 (Java & JS & Python) qfc_128220的博客 3995

2022.Q4 已支持Java、JavaScript、Python，深度优先搜索DFS

华为机试真题-C++实现【处理器问题】 【2022.11 Q4 新题】 MISAYAONE的博客 1246

某公司研发了一款高性能AI处理器。每台物理设备具备8颗AI处理器，编号分别为0、1、2、3、4、5、6、7。编号0-3的处理器处于同一...

华为od机试题2 真题_华为od机试题库及答案_chenzm666666的博客-CSDN... 3-10

华为od机试真题 77.满足最大消费额度 76.小朋友身高位置 75.字符串连续出现最大次数 74.最少停车数 73.字母多条件排序 71.交叉排序...

【华为OD机试真题 Python OD笔试题】处理器问题： 某公司研发了一款高性能AI处理器。每台物理设备具备8颗AI处理器，编号分别为0、1、2、3...

华为机试真题 Python 实现【处理器问题】 MISAYAONE的博客 6238

某公司研发了一款高性能AI处理器。每台物理设备具备8颗AI处理器，编号分别为0、1、2、3、4、5、6、7。编号0-3的处理器处于同一...

#面试经验001 华为/字节/滴滴/中兴/美团 AI算法题汇总 (含答案) m0_49629753的博客 679

1、怎么解决过拟合 正则化 (L1、L2正则化) 数据增强 (数据集越多,过拟合的概率越小,可以通过旋转、缩放、剪切、添加噪声等操...

AI处理器技术发展 luogantao的博客 108

本文为个人在本科阶段所修课程《计算机系统结构》中提交的期末论文，主要涉及了AI处理器中GPU和DianNao系列芯片的粗浅介绍。A...

[华为机试]某公司研发了一款高性能AI处理器 SCUhzs 523

[华为机试]某公司研发了一款高性能AI处理器

【华为机试真题Java】最远足迹 171

某探测队负责对地下洞穴进行探险。探险队员在进行探险任务时，随身携带的记录仪会不定期地记录自身的坐标，但在记录的间隔中...

放大招！华为发布算力最强AI处理器及全场景AI计算框架 华为云官方博客 2259

6月23日，华为发布算力最强AI处理器Ascend 910（昇腾910）及全场景AI计算框架MindSpore。华为公司轮值董事长徐直军表示...

2019新书《昇腾AI处理器架构与编程》思维导图。附件是FreeMind源文件，供大家编辑丰富。昇腾AI处理器架构与编程... 3245

【摘要】 2019新书《昇腾AI处理器架构与编程》大綱思维导图。附件是FreeMind源文件，供大家编辑丰富。昇腾AI处理器架构与编程... 1301

《谈一本好书》《昇腾AI处理器架构与编程》--神经网络基础知识(2) 华为云官方博客

1卷积神经网络：输入层之前提到多层感知机的参数太多，导致训练耗时并且对图像处理也不具有优势，因此大神们就提出了多层神...

“相关推荐”对你有帮助么？

👍 非常有帮助 👎 没帮助 😐 一般 😊 有帮助 😄 非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公司地址：北京市海淀区中关村大街18号C座15层 邮编：100048 业务联系：010-63171895 广告合作：010-63171895 内容合作：010-63171895

若博豆 已关注 1 6 1 专栏目录 已订阅