2015“亚马逊•种子杯”

决赛试题

**2015/11/14**

## 题目概述

电商公司A每天要处理X种货物，每种包含长、宽、高、件数信息{length, width, height, units}，电商公司A提供了Y个容器来打包这些货物，每个容器包括长、宽、高、最多容纳货物种类信息，如{length, width, height, max num of ASINs}。

不同的打包算法，容器空间的利用率也不同，好的打包算法能更充分地利用容器的空间，节省库存资源和人力资源。

## 二、题目要求

我们要新建一个库房，库房中有许多空的容器，我们需要将一批货物全部装入这些容器，并尽可能地充分利用容器的空间，缩小公司的库存成本。

具体要求如下：

设计打包算法，将这批货物全部放入容器中，并使容器的存放效率尽可能地高，且让程序的运行时间尽可能地短。

**提示：存放效率=所有货物体积/所有放入货物的容器体积**

## 详细需求

### 容器及货物信息

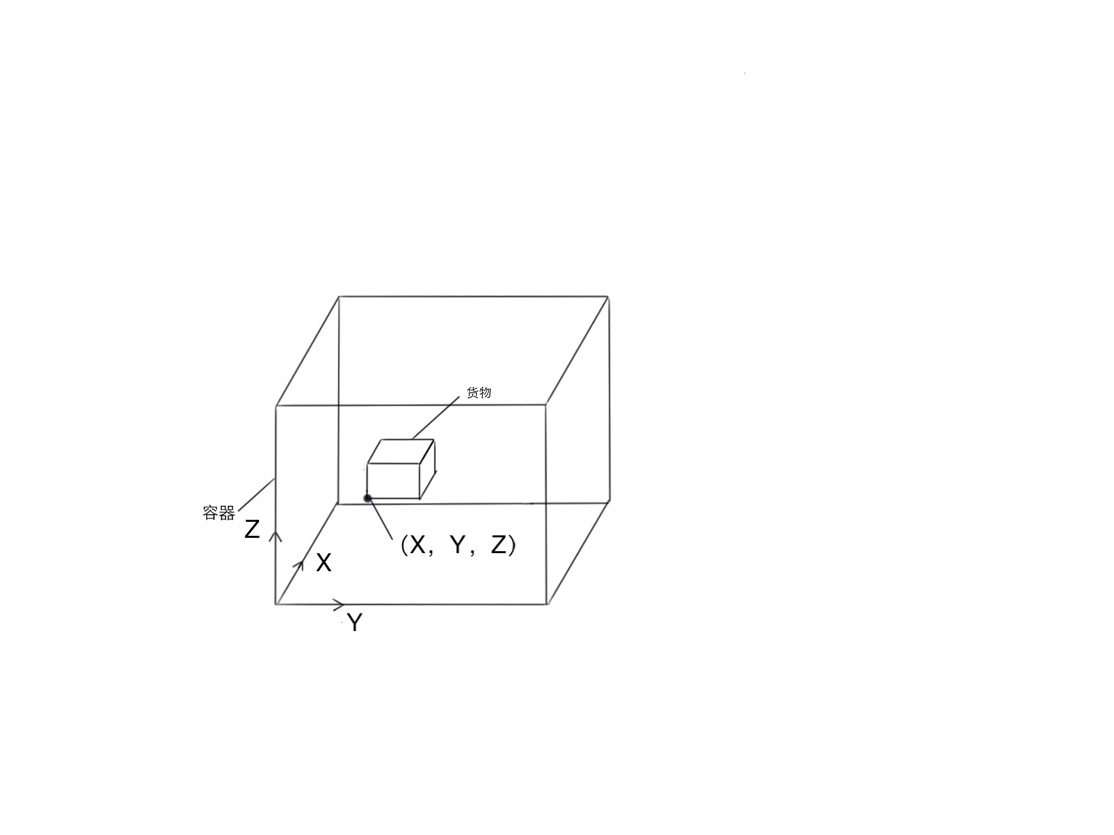
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 货物种类 | 长 | 宽 | 高 |  |  |
|  |  |
| A | 2 | 3 | 4 |  |  |
| B | 3 | 4 | 5 |  |  |
| C | 4 | 5 | 6 |  |  |
| D | 5 | 6 | 7 |  |  |
| E | 6 | 7 | 8 |  |  |
| F | 7 | 8 | 9 |  |  |
| G | 8 | 9 | 10 |  |  |
| H | 9 | 10 | 11 |  |  |
| I | 10 | 11 | 12 |  |  |
| J | 11 | 12 | 13 |  |  |
|  |  |  |  |  |  |
| 容器种类 | 长 | 宽 | 高 | 最多存放物品种类 | |
|
| X | 11 | 13 | 17 | 6 | |
| Y | 16 | 18 | 22 | 7 | |
| Z | 21 | 23 | 27 | 8 | |
| M | 26 | 28 | 32 | 9 | |
| N | 31 | 33 | 37 | 10 | |
|  |  |  |  |  |  |
| 数量 规模 | X | Y | Z | M | N |
|
| 8 | 5 | 5 | 5 | 5 | 5 |
| 10 | 5 | 5 | 5 | 5 | 5 |
| 80 | 23 | 17 | 13 | 17 | 23 |
| 100 | 23 | 17 | 13 | 17 | 23 |
| 800 | 137 | 177 | 153 | 163 | 134 |
| 1000 | 137 | 177 | 153 | 163 | 134 |
| 4000 | 732 | 712 | 775 | 781 | 542 |
| 5000 | 732 | 712 | 775 | 781 | 542 |
| 8000 | 1423 | 1498 | 1467 | 1598 | 1245 |
| 10000 | 1423 | 1498 | 1467 | 1598 | 1245 |

**提示：“规模”指A、B、C、D、E、F、G、H、I、J需要装填的每种货物的数量；“数量”指当前仓库内可用的X、Y、Z、M、N每种容器的数量；容器可有剩余但货物必须装填完。**

### 笛卡尔坐标系

我们使用笛卡尔坐标系来描述货物在容器中的位置，**货物在容器内允许横放、竖放和侧放，但不允许斜放**；以容器底部的顶点为原点，**容器的X、Y、Z轴按照题目表格中给定的容器长宽高方向建立，不允许改变（如图）**，使用货物距离原点最近的一个顶点（即左下顶点）和货物的长、宽、高来确定货物的位置。

如：某一个货物的位置是(A1,(1,1,1))，代表该货物离原点最近的一个点（即左下顶点）是(1,1,1)，该货物的id是A1。



**提示：X、Y、Z轴分别代表容器的长、宽、高所在的方向**

### 输入输出规范

最终生成的SeedCup.exe需要完成以下的任务：

输入对应的规模数字来解析同目录下的对应的box.txt来获得容器信息；解析同目录下的对应的pro.txt来获得货物信息。

示例：box2.txt

X 11 13 17 6 23

Y 16 18 22 7 17

Z 21 23 27 8 13

M 26 28 32 9 17

N 31 33 37 10 23

**说明：第一个参数表示容器种类；第2，3，4个参数分别表示该种容器的长，宽，高；第五个参数表示该种容器最多存储货物的种类数；第6个参数表示可用容器数量；每个参数之间用空格隔开**

示例：pro2.txt

A 2 3 4 100

B 3 4 5 100

C 4 5 6 100

D 5 6 7 100

E 6 7 8 100

F 7 8 9 100

G 8 9 10 100

H 9 10 11 100

I 10 11 12 100

J 11 12 13 100

**说明：第1个参数表示货物种类；第2，3，4个参数分别表示该种货物的长，宽，高；第5个参数表示货物数量；每个参数之间用空格隔开**

输出：根据你的移动算法，输出ans1.txt——ans10.txt来存储各个容器中的货物坐标以及移动后容器的存放效率。

示例：ans.txt

A1 X1 0 0 0 2 3 4

B1 Y1 0 0 0 3 4 5

C1 Z1 0 0 0 4 5 6

50%

**说明：第1个参数表示货物的种类和ID（ID为大于等于1的整数，请顺序命名）；第2个参数表示容器的种类和ID（ID为大于等于1的整数，请顺序命名）；第3，4，5个参数表示容器中该货物的存放坐标；第6，7，8个参数表示容器中该货物摆放时的x，y，z轴（2 3 4即表示以货物长的方向为x轴，宽的方向为y轴，高的方向为z轴摆放）；每个参数之间用空格隔开**

**程序运行，命令行等待用户输入数据规模i，读入数据规模i后，程序需要读取input文件夹下的proi.txt，boxi.txt文件，输出ansi.txt到output目录，程序就结束。**

**eg:**

**程序运行，命令行等待用户输入数据规模i，读入数据规模1后，程序需要读取input文件夹下的pro1.txt，box1.txt文件，输出ans1.txt到output目录，程序就结束。**

## 四、作品提交

比赛终止时间为2015/11/15 12：30；答辩ppt下午3：00前提交

目录说明如下:Src 文件夹放置源代码及工程文件,Bin 文件夹放置最后生成的可执行文件, Doc 文件夹放置说明文档,Test 文件夹放置输入文件

## 五、评分说明

**决赛的测试用例的规模分为8、10、80、100、800、1000、4000、5000、8000、10000共10个级别，每个级别的分数由两部分组成：（1）存储效率得分 （2）程序运行时间得分。总的得分为参赛队伍在五个级别的分数之和。**

|  |  |  |  |
| --- | --- | --- | --- |
| **规模** | **总分** | **存储效率（分）** | **运行时间（分）** |
| **8** | **5** | **4** | **1** |
| **10** | **5** | **4** | **1** |
| **80** | **10** | **8** | **2** |
| **100** | **10** | **8** | **2** |
| **800** | **10** | **8** | **2** |
| **1000** | **10** | **8** | **2** |
| **4000** | **10** | **8** | **2** |
| **5000** | **10** | **8** | **2** |
| **8000** | **15** | **12** | **3** |
| **10000** | **15** | **12** | **3** |

#### 存储效率评分标准：

**采用相对分数来计分，达到最大存储效率的队伍获得满分，其他队伍的得到为 ： 满分\*（存储效率/第一名的存储效率）**

**例如：在规模为10的级别上，队伍A的存储效率最大，假设达到 80%，则该队伍获得8分，队伍B的存储效率为60%，则队伍B的得分为8\*（60%/80%）=6 分。**

#### 运行时间评分标准：

**按照时间排名来计分，运行时间最短的队伍获得满分，第二名至最后一名的分数依次为90%，80%，70%，60%等依次递减。**

**例如：在规模为10的级别上，队伍A的运行时间最短，则队伍A的得分为2分，第二名及之后的队伍的得分依次为 2\*90%，2\*80%，2\*70%等等。**

**（提示：运行时间最长不得超过20 min,否则该项得分为0）**