2024 华为软件精英挑战赛

初赛任务书

文档版本 01

发布日期 2024-03-07

华为技术有限公司

目 录

1 更新记录

表1-1 更新记录

版本	修改说明	发布时间
01	第一次正式发布。	2024-03-07

2 背景信息

智慧港口以信息物理系统为框架，通过高新技术的创新应用，使物流供给方和需求方的沟通融入“集、疏、运”一体化系统，极大的提升了港口及相关物流园区对信息的综合处理能力和对资源的优化配置能力。

智慧港口以现代化基础设施设备为基础，以云计算、大数据、物联网、移动互联网、智能控制等新一代信息技术与港口运输业务深度融合为核心，以港口运输组织服务创新为动力，以完善的体制机制、法律法规、标准规范、发展政策为保障，能够在更高层面上实现港口资源优化配置，在更高境界上满足多层次、敏捷化、高品质港口运输服务要求的，具有“生产智能、管理智慧、服务柔性、保障有力”等鲜明特征的现代港口运输新业态。智能监管、智能服务、自动装卸是智慧港口的主要呈现形式。

图2-1 智慧港口

在智慧港口领域，如何规划多机器人的任务执行，以实现最优调度；如何控制泊位和机器人的配合，达到最高价值等都是非常有价值的算法难题。本次比赛通过软件模拟了智慧港口的状态信息，由选手来挑战这些有价值的问题。

期待您的精彩解决方案。

3 题目定义

3.1 题目概述

题目概述

- **目标**

  赚取更多的资金。

- **程序操控方式**

  选手作为运输公司来运输货物赚取资金，每个选手有 5 艘轮船、10 个机器人。选手需要使用机器人来执行移动、搬运等动作来完成物品递送任务，同时赚取利润。在运行结束时，选手拥有的资金数即为最终分数，所获得的资金越高越好。

  初赛时间为 **15,000 帧**（最多 **5 分钟**）。

- **程序交付方式**

  选手程序通过标准输入和标准输出与判题器进行交互。判题器运行帧率为每秒 50 帧，对于每一帧，判题器都会把场上的实时信息通过标准输入传递给选手程序，同时从选手程序读取机器人的操控指令作用到各个机器人上。每一帧有 1000/50=20ms 的时间，由于判题器需保留 5ms 执行计算来模拟真实场景，故选手程序需要在 15ms 内做出每一帧的决策，如果超过 15ms 未做出决策，则系统将直接忽略这一帧的控制进入下一帧，并且在选手程序返回控制指令前，不会再发送状态数据给程序。

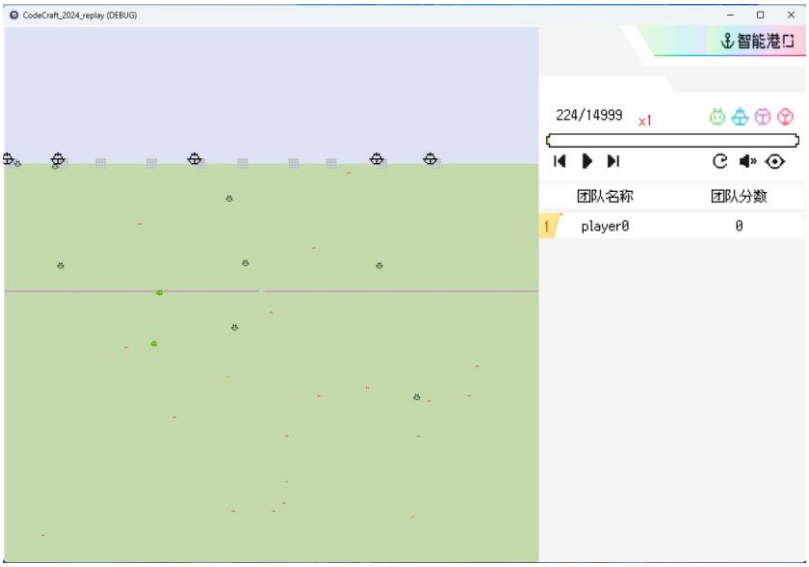
  注意，你不需要让自己的程序具备处理 50FPS 的性能，程序处理帧率低于 50FPS 也能正常运行（例如只处理 10FPS 也可以），但是处理更高的帧率可以让你实现更高精度的控制。

  程序的输入和输出格式请参考 3.4 输入与输出格式。

- **判题器使用**

  今年的比赛判题器与数据集完全开放给大家下载，并且做了跨平台设计（Windows/Linux/MacOS），大家可以根据自己习惯选择对应版本下载。但是请注意，比赛平台使用 Linux，因此无论你选择何种平台开发调试，都必须确保你的代码可以在 Linux 下编译运行。

  运行判题器中的 run_simple_demo 可快速运行一个 DEMO，运行界面如下：



术语

表3-1 术语

名词	解释
地图	地图是一个 200*200 的封闭区域。
坐标系	往下为 X 轴正方向，往右为 Y 轴正方向。地图左上角坐标为原点(0,0)，右下角坐标为(199,199)。
物品	物品随机生成在地图上，可由机器人在一处物品处搬运之后放置在泊位上可供船舶运输产生利润。
轮船	轮船是一个 2*4 的矩形，轮船有一定的容积，需要停留在泊位上，可以把泊位上的货物运走。在初赛中，不需要考虑轮船的路径规划问题，只考虑轮船放置在哪个泊位，以及何时运输即可。
泊位	泊位是一个 4*4 的矩形，分为海岸和陆地部分。泊位的陆地部分可以放置货物，泊位的海岸部分可以放轮船。设泊位的左上角坐标为(x,y)，该泊位的右下角坐标为(x+3,y+3)。每个泊位有运送货物到船上的效率和轮船运走货物产生收益的时间，在同一局游戏中，泊位信息不会发生变化。特殊的，在初赛阶段不特别区分泊位的海岸和陆地部分，机器人可以移动到泊位上，无需考虑和船的碰撞。
机器人	机器人是一个占据 1*1 格子的物体，可携带一个物品。由选手程序进行操控，可以执行**上下左右**四种移动操作，当机器人位于物品生成处时，可以搬运物品，当机器人位于泊位处时，可以放下物品。
运输	指轮船从泊位驶出运输货物产生价值。价值会在运输到达虚拟点瞬间立即产生（同一帧），选手也可以在运输到达虚拟点帧的时刻立即命令轮船移动到一个泊位上（同一帧）。

3.2 运输公司机制

3.2.1 物品和售价机制

物品将会随机生成在地图的每一个区域，物品有着不同的价值，每个物品会在指定位置停留 **20s(1000 帧)**的时间，若生成后 1000 帧内未被搬运则在地图上消失，选手可以操控机器人将物品从生成位置运送到泊位，由泊位装卸货物到轮船，最后由轮船运输至虚拟点产生价值。获得资金会在轮船运输至虚拟点的帧时刻完成。

3.2.2 虚拟点和运输机制

初赛阶段，不考虑海面上的移动路径，为此在海上设置一个虚拟点。轮船需要运输货物到虚拟点，每个泊位会有移动到虚拟点的时间(虚拟点移动到泊位的时间同)，当轮船运送货物到虚拟点的瞬间产生价值，同帧轮船可以执行 move 指令到新泊位。

3.2.3 轮船和泊位机制

初赛阶段，选手有 5 艘轮船可以操控，我们假定轮船可以直接停靠在泊位上，所以不需要选手考虑轮船的路径规划。在初赛阶段，所有的海面上（包括泊位间的移动和运输）操作路径均不考虑，船在移动状态时均可以看作在地图上消失。

泊位上有装卸机器，可以将泊位上的货物装载到轮船上。每个泊位有不同的装卸速度，每个泊位只能容纳一艘船。轮船处于泊位时，装卸机器将会将泊位上放置的货物运到轮船上。轮船有容积，轮船装满后将不能再装载货物到轮船上。泊位的装卸机器会按照选手放置物品在泊位上的顺序进行装载。

轮船和泊位之间有到达、装载、驶离三种动作。每一帧在泊位处将依次执行到达、驶离、装载三种动作。

当两个轮船在同一帧到达同一泊位时，先输入指令的轮船先进入泊位。对于另一艘轮船，选手可以选择不输入指令，继续在泊位等待，当泊位空闲时到达泊位，也可以选择输入移动、运输指令，移动到其他泊位或者移动到虚拟点。

由于不考虑轮船路径规划，我们假定泊位间的移动时间都是 **500 帧(10s)**，判题器将告知选手轮船从每个泊位运输产生价值的时间。选手可以随时操控轮船进行运输或移位。轮船结束运输产生价值的帧时刻，选手可以执行 move 指令把轮船从虚拟点移动到一个空闲的泊位上。

3.2.4 机器人机制

选手有 10 个机器人可以操控，机器人有上下左右四个方向的移动，每帧可以移动一格，如果机器人碰撞到了墙壁或海（即机器人的移动目标位置和墙壁重合）或者两个机器人相互碰撞（即两个机器人的目标位置重合），则会停在原地（未触发该机器人当前帧动作，不进行移动及取货，并有可能产生连锁反应）**20 帧(400ms)**的时间。在机器人和物品坐标重合的帧时间可以使用 get 命令获取该坐标的物品，当机器人处于泊位范围的帧时间，可以使用 pull 命令将机器人身上的物品放置在泊位上。

机器人每帧的动作可以分为移动前动作，移动，移动后动作三个部分。每个帧时刻将同时结算所有机器人的移动前动作，然后继续同时结算所有机器人移动动作、最后同时结算移动后动作。若移动动作产生碰撞，则机器人的一帧的所有动作均失效。若未进行移动动作，所有该帧的动作视为移动前动作。

如机器人 A 位于(1,1)进行取货 1，移动到(1,2)，取货 2 动作，机器人 B 位于(2,2)进行移动到(1,2)动作，机器人 C 位于(1,0)进行移动到(1,1)动作，可以发现机器人 A 和机器人 B 的移动目标位置相同，机器人 C 移动到了原机器人 A 的位置，机器人 A 和机器人 B 发生碰撞，返回原位后，机器人 C 和机器人 A 发生碰撞，机器人 C 也回到原位(1,0)，三个机器人同时处于恢复状态，机器人 A 的取货 1、取货 2 操作均失败。

3.3 帧的行为结算顺序

在一帧中，会按照如下顺序结算以下行为：

1. 机器人恢复。
2. 船舶到达、进入港口。
3. 物品生成。
4. 生成场面信息，输出给选手。
5. 读取选手指令。
6. 执行机器人指令。
7. 执行船舶指令。
8. 港口装卸货物。

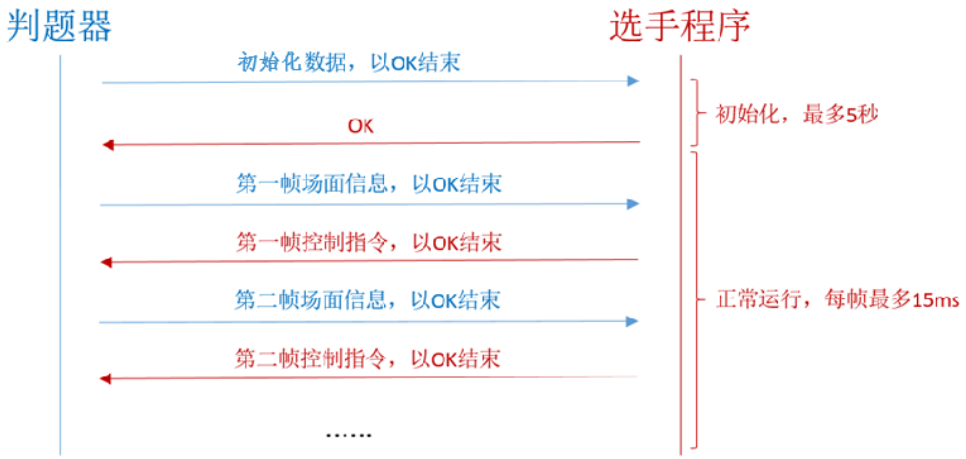
需要特别注意的是，船舶的移动指令会立即生效，导致船舶离开港口。在船舶指令和港口装卸货物结算之间没有船舶到达的结算。因此在结算港口装卸货物时，可能会出现有船舶在港口外等待，但港口内没有船舶的情况

3.4 输入与输出格式

3.4.1 选手程序和判题器交互过程

选手程序和判题器的交付过程如图 3-1 所示。

图3-1 程序和判题器的交付过程



3.4.2 初始化数据格式介绍

地图

地图数据（可参考 maps/*.txt）是一个 200 行 200 列的字符矩阵。地图数据中每个字符含义如下：

```
'.' : 空地
'*' : 海洋
'#' : 障碍
'A' : 机器人起始位置，总共 10 个。
'B' : 大小为 4*4，表示泊位的位置,泊位标号在后泊位处初始化。
```

须知

- 保证机器人初始位置不会重叠。
- 保证泊位不会重叠。

泊位

泊位数据是由 10 行 5 列数字组成。

每一行五个整数 id, x, y, time, velocity 表示一个泊位。id(0 <= id < 10)为该泊位的唯一标号，(x,y)表示该泊位的左上角坐标,time(1 <= time <= 1000)表示该泊位轮船运输到虚拟点的时间(虚拟点移动到泊位的时间同)，即产生价值的时间，时间用帧数表示。Velocity(1 <= Velocity <= 5)表示该泊位的装载速度，即每帧可以装载的物品数，单位是：个。保证对于每次提交全部泊位到虚拟点时间和相同。

轮船

一行一个整数 capacity(1 <= capacity <= 1000)，表示船的容积，即最多能装的物品数。

3.4.3 输入格式

所有数据采用文本格式通过标准输入和标准输出进行交互，数值之间用空格分隔。

- **初始化：**选手程序初始化时，将输入 200 行*200 列的字符组成的地图数据、10 行的泊位数据和 1 行船的容积，然后紧接着一行 OK。
- **每一帧交互：**
  - 第一行输入 2 个整数，表示帧序号（从 1 开始递增）、当前金钱数。
  - 第二行输入 1 个整数，表示场上新增货物的数量 K（0<=K<=10）。
  - 紧接着 K 行数据，每一行表示一个新增货物，分别由如下所示的数据构成，共计 3 个数字。

名称	数据类型	说明
坐标	2 个整数 x,y	该货物的坐标
金额	正整数	该货物的金额(<= 1000)

  - 接下来的 10 行数据，每一行表示一个机器人，分别由如下表格中所示的数据构成，每行 4 个数字。

名称	数据类型	说明
是否携带物品	整数	• 0 表示未携带物品。 1 表示携带物品。
坐标	2 个整数 x,y	该机器人的坐标
状态	整数	• 0 表示恢复状态 1 表示正常运行状态

  - 接下来的 5 行数据，每一行表示一艘船，分别由如下表格中所示的数据构成，每行两个数字。第 i 行表示标号为 i-1 的轮船。

名称	数据类型	说明
状态	整数	• 0 表示移动(运输)中 • 1 表示正常运行状态(即装货状态或运输完成状态) 2 表示泊位外等待状态
泊位 ID	整数	表示目标泊位，如果目标泊位是虚拟点，则为-1

    最后，判题器会输出一行 OK，表示所有数据已经写入完毕。
  - 示例：合法的一帧输入可能是这样的

```
114 199346
2
```



```
1 5 20
5 7 30
0 1 2 1
1 100 150 1
1 95 174 1
0 14 156 1
0 154 47 0
1 17 41 0
1 99 152 1
1 92 175 1
0 13 152 1
0 152 41 0
1 16 73 0
1 2
1 1
0 6
1 -1
0 8
```

- **判题结束：**

  判题结束时，判题器会关闭输入管道，同时选手程序会读到 EOF(end of file)，此时应当退出程序。

3.4.4 输出格式

- **初始化：**读入地图数据并完成初始化后，选手程序应当输出一行 OK，告诉判题器已就绪。
- **每一帧交互：**
  - 先进行机器人指令的输出，每行一个指令，按照 "指令 <机器人 ID> [参数 2]"的格式进行输出，机器人 ID 的取值范围是[0,9]，对应机器人的输入顺序。支持的指令如下所示。

指令	参数 1	参数 2	说明
move	机器人 ID 取值[0, 9]	[0,3]之间的整数	• 0 表示右移一格 • 1 表示左移一格 • 2 表示上移一格 3 表示下移一格
get	机器人 ID 取值[0, 9]	无	如机器人在货物生成处，并处于未携带物品状态，则取货成功。
pull	机器人 ID 取值[0, 9]	无	如机器人在泊位处，并处于携带物品状态，则放置成功。

  - 紧接着，每行一个指令，按照"指令 <船 ID> [参数 2]"的格式进行输出，船 ID 的取值范围是[0,4]，对应船的输入顺序。支持的指令如下所示。

指令	参数 1	参数 2	说明
ship	船 ID 取值[0, 4]	泊位 id 取值[0, 9]	表示船移动到泊位 ID。
go	船 ID 取值[0, 4]	无	表示船从泊位驶出至虚拟点运输货物。

说明

如果轮船已经移动到虚拟点，可以在同一帧使用 ship 指令让其从虚拟点移动到泊位，如果该轮船上次指令时间未结束再次收到指令，会以新指令为准重新计算时长。

ship 的目标如果是当前港口，会作为一次耗时 1 帧的移动处理。

同一个机器人和轮船可以在同一帧内执行多条指令，因此，机器人可以同一帧内移动后立刻取货，同一船只可以到达泊位的同一帧立刻装货。

- 当你输出完所有指令后，紧跟一行 OK，表示输出结束。例如，一个可能的输出是：

```
1140
move 1 2
get 1
ship 3 2
go 2
OK
```

说明

大多数语言会默认对输出数据做缓冲，因此在输出完一帧控制数据时，你应该主动 **flush 标准输出以避免判题器读不到数据，导致超时**。此外，由于标准输出已用于比赛交互，因此不要往标准输出打日志等其他内容，以免造成命令解析错误。平台上没有任何写文件权限，故正式提交版本不要写日志文件，你可以使用 stderr 将日志输出到控制台以方便调试。

特别说明

当帧数为 1 的时候，给出机器人位置为机器人的初始位置，给出的轮船位置是无效的，选手不用考虑轮船的初始位置，轮船的初始位置均认为在虚拟点，即指令为 1 -1，后续的轮船位置改变需要考虑轮船移动时间。

3.5 异常判定与处理

- **程序异常：**判为 0 分。

  包括：程序崩溃、输出格式错误、指定机器人下标越界、指定船下标越界、指定泊位下标越界、输出数据量超长（1 帧内超 8KB）。
- **逻辑异常及参数错误的命令：**忽略该指令，不扣分。

  在无法执行某个指令的时候执行了某个指令，该指令被忽略，不扣分。例如，无法取货、送货的时候执行了对应指令。
- **初始化响应超时：**直接进入比赛

5 秒内未收到选手的 OK，则直接进入比赛，并且直到收到选手的 OK 之前，不会给选手下发数据。

- **控制帧响应超时**：忽略该帧控制指令

  选手程序没有在 15ms 内返回控制指令，则忽略该帧的控制行为（即：跳帧），直到收到控制指令之前，不会继续给选手程序下发数据。

须知

跳帧后收到的控制数据会作用在当前最新的一帧，例如第 1000 帧下发的数据，选手程序直到 1010 帧才做出响应，那么该控制指令会作用于 1010 帧，此时有可能货物消失，机器人已经不在原来位置，从而导致一些取货、送货指令失败（失败的指令被忽略）。

