

- [参赛须知](#)
[竞赛题目](#)
[测评方案](#)
[提交说明](#)
[比赛规则](#)
[代码规范](#)
[作品上传](#)

参赛须知

文档（注册之后可以下载）	格式	下载
case1.zip	.zip	

竞赛题目

2035年，随着火星基地的建立，高效清洁的新型能源逐渐成为人类在火星生存的关键性因素。星际能源署发现火星上存在一种稀有的 α 晶体，能够释放巨大的能量。火星基地遵循《星际资源的和平探索利用条约》，派遣航空飞行器编队勘探 α 晶体。



为了更好地规划 α 晶体资源的探测路线，勘探编队结合火星地形，预估了每个可能存在 α 晶体的勘探区域，以及探测该区域所需的时间。时间紧迫，勘探编队决定派遣3架航空飞行器同时对可能存在 α 晶体的区域进行勘探。

3架航空飞行器分别从不同的区域出发开始勘探工作，勘探完毕后统一在指定区域汇合。请编写程序规划每架航空飞行器的勘探路线。



要求：

- 必须覆盖全部区域（即每个区域确保至少有一架航空飞行器进行过探测）；
- 耗时最长的航空飞行器和耗时最短的航空飞行器时间差不能超过耗时最短航空飞行器用时的30%。

在此基础上，三架航空飞行器勘探的任务尽量均匀分配，不要重复，且总路径长度要尽可能短，请给出你认为的最佳规划方案。

测评方案

输出描述

输出三行，分别给出每个队伍（依次0号、1号、2号）的规划路线。每行为一队的数据，是该队途径的点，行内数据之间要用英文半角分隔。

输出示例

```
0 12 13 12 6 11 10 19
1 15 3 9 17 16 17 9 10 19
2 7 5 4 8 18 14 11 10 19
```

按照以上示例，0队的路径是0、12、13、12、6、11、10、19，路径长度为8，勘探耗时为7天（7x24小时）。

我们将从以下几点评价路线规划方案：（替换：包含权重的）

指标	备注
各航空飞行器均衡程度	各队伍耗时
航空飞行器勘探单条线路最长路径	
重复途径点数	
代码规范性	使用Google的cpplint检测文件，设置等级为3，输出的error
占用内存	
运行时间	
题目解析	

代码示例

[case1.zip](#)，目录结构为

```
input.txt
output.txt
case1.cpp
```

其中的txt文件进作为调试、参考之用，提交时只提交case1.cpp代码文件。

提交说明

源代码：只需提交一个utf8编码的cpp文件，命名为`"case1.cpp"`，参见示例代码，文件大小不得高于5M。

解题思路：提交名为case1.pdf的文件，文件大小不得高于20M，文件内容无固定范本，思路清晰条理即可。

比赛规则

- 请确保代码主要原创。
- 本次比赛的提交的代码需要能够在 g++ (GCC) 9.3.0 Linux/amd64，并基于C++11标准下正常编译并运行，即该命令能成功编译 `g++ std=c++11 case1.cpp`。
- 程序统一运行在单核单线程1Ghz的CPU和256M内存的容器内，OOM或者运行时间超过1分钟将判定为程序无效。
- 开发调试阶段，可使用示例中提供的数据进行调试。

代码规范

- 输入文件：作为第一个参数提供文件路径。
- 输出文件：作为第二个参数提供文件路径。
- 代码标准：所有依赖的函数必须明确地在源文件中 `#include <xxx>`，不能通过IDE的工程设置而省略常用头文件；代码中只允许使用C++ 11标准库（<https://zh.cppreference.com/w/cpp/header>），不允许依赖其它第三方代码，不可使用C++14、C++17和C++20。
- 评估标准中，代码规范会参考Google的cpplint（<https://github.com/google/styleguide/tree/gh-pages/cpplint>），等级为3，作为做该项打分依据，命令为 `./cpplint.py -verbose=3 case1.cpp`，执行前，会把代码中所有的NOLINT注释清除，保证评分的公允性，示代码目前0个error。

```
$ ./cpplint.py --verbose=3 xx.cpp
xx.cpp:8: Do not use namespace using-directives. Use using-declarations instead. [build/namespaces] [5]
Done processing xx.cpp
Total errors found: 1
```

- 注释代码，函数、方法、属性、常量都要注明其作用，以及复杂的逻辑也能做到引导性解释。
- main函数return 0。
- 使用示例代码中已预处理好文件读写（为规避OOM，如有能力可尝试优化I/O），请实现其中的YourFunction方法，其中参数input_lines为原始数据文件行的集合，output_lines为结果集合，请按照提干要求将结果以每行作为一个元素放入output_lines集合中。

```
// 你需要实现代码实现的函数
void YourFunction(vector<string> &input_lines, vector<string> &output_lines)
{
    // 请实现你的代码，将结果按行写到output_lines中
}
```

Copyright @ 2012-2021 全国算法设计挑战赛组委会 保留一切权利 京ICP备11018333号-11