

>

接口

1. 预测函数

```
//需要预测的数据
int vmTotalNum = 0; //虚拟机总数
int[] vmsNum = new int[15]; //不同规格虚拟机数量，初赛最对需要前15种，vmsNum[i-1]表示第i种规格的虚拟机数量

//函数返回类型是void，需要预测的数据都定义为类的全局变量
public void predict(int[][] trainData, int day_span,int vmClassNum){

    /*参数说明
        *int[][] trainData: 大小为int[days][15],trainData[i][j]表示第i天的第j种规格虚拟机数量
        *int day_span: 时间跨度，day_span=i,表示预测之后i天的数据
        *int vmClassNum: 需要预测的虚拟机规格数量，vmClassNum=2，表示需要预测2种规格的虚拟机
    */

    //do your work*****

}
```

2. 放置函数

```
//要求输出的数据
int serverNum = 0; //物理服务器数量
int[][] serversDetail; //每个物理服务器的放置虚拟机的具体信息，数组大小需要根据预测结果serverNum确定。得出serverNum后再初始化
//大小为int[serverNum][15]。serverDetail[i][j]表示物理服务器i上放置的规格为j的虚拟机数量

//函数返回类型是void，需要输出的数据都定义为类的全局变量
public void arrange(int[] cpuInfo, int target, int[] vmsNum, int[][] vmClassSpecs, int[] vmsNum){

    /*参数说明
        *int[] cpuInfo: 大小为int[2]，[物理服务器CPU核数 内存大小（GB）]
        *int target: 需要优化的对象，target=1: 是cpu; target=2, 是内存
        *int[] vmsNum:不同规格虚拟机数量，大小是int[15]。初赛最对需要前15种，vmsNum[i-1]表示第i种规格的虚拟机数量
        *int[][] vmClassSpecs: 需要预测的虚拟机规格对应的参数，大小是int[vmClassNum][3]
        * vmClassSpecs[i][0]表示规格名称，vmClassSpecs[i][0]=3表示规格名称是3
        * vmClassSpecs[i][1]表示规格vmClassSpecs[i][0]的cpu核数
        * vmClassSpecs[i][2]表示规格vmClassSpecs[i][0]的内存大小
    */

    //do your work*****

}
```

3.数据

- data_total_12345_array.txt是2015年1~5月的数据，data_total_121_array.txt是2015年12月和2016年1月的数据。每一行是一天的数据，每一列对应一种规格。比如第一行第一列是2015年1月1日的规格1的虚拟机数量。
- 练习数据.zip里面是官方提供的原始资料。