车站信息

StationReference = Name | MayBeInitial | Platforms

名称	是否可初始化?	站台
a = 汉口动车所	1	1, 2
b = 丹水池联络线丹水池方向(外侧)	1	1, 2
c = 沪蓉线南京南方向(内侧)	1	1, 2
d = 汉孝城际孝感东方向	1	1, 2
e = 汉口站	1	1到 18
f = 汉西联络线汉西方向(最外侧)	1	1, 2
g = 沪蓉线成都东方向(中间)	1	1, 2
h = 汉丹线丹江口方向(最内侧)	1	1, 2
i = 客整所12/机务折返段3	1	1, 2, 3

时刻表

Format: ReportingNumber TrainType MaxSpeedKmph TrainComposition Flags : StationVisit1 StationVisit2 ...

StationVisit format:
StationReferencePlatformNumberFromDurationMinutes

TrainType format:
COMMUTER | FREIGHT | IC | URBAN

TrainComposition format:
vvv...
Each v represents one vehicle. L = locomotive (or control post), C = cargo car, P = passenger car

Flags format:
ff
Each f is one flag. 0 = flag not set, 1 = flag set, X = position not used
Flag positions:
1 unused (X)
2 NoBrakingPenalization - if set (1), train does NOT receive penalization when braking at signals

例子

列车编号	类型	最高时速	动拖布置	flag	车站1	车站2	车站3
G3472	COMMUTER	300	LPPLLPPL	X1 :	b#2#21:40:00#0	e#0#21:45:00#30	a#0#22:19:00#0

车次相关信息

车次信息不变 默认commuter

依照类型设置最高速度 D->200 G->300....

车辆细节 先默认 MTTM 后续修改长编或者重连?

停站相关信息

仅三点式 进场车站 停站 离场车站

始发终到则进行利用抵达时刻等替换

公式=IF(B2<>J2,B2,TEXT(E2-0.5/24,"h:mm:ss"))

车站 筛选所有车次始发终到，设置字典更改进场离场

股道指定 随机指定?

表格信息

从路路通车站时刻表截图ocr导出为Excel进行处理

	A	B	C	D	E
1	车次	到时	开时	始发站	终到站
2	Z4	23:09:00	23:33:00	重庆北	北京西
3	Z96	23:15:00	23:40:00	重庆西	北京西
4	D5810	19:45:00	20:00:00	宜昌东	汉口
5	D3011	19:56:00	20:26:00	上海虹桥	汉口
6	D2278	19:57:00	20:27:00	重庆北	汉口
7	G6784	20:18:00	20:48:00	巴东	汉口
8	D5986	20:19:00	20:49:00	宜昌东	汉口
9	D3033	20:37:00	21:07:00	上海虹桥	汉口
10	D5964	20:40:00	21:10:00	宜昌东	汉口

从路路通车站车站大屏截图ocr导出为Excel进行处理

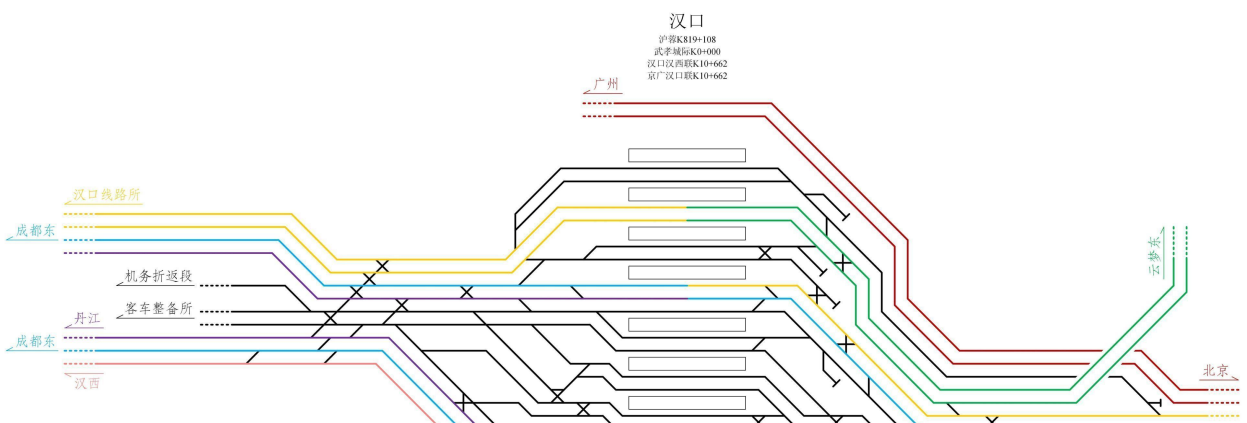
	A	B	C
1	车次	检票口	
2	D2158	11	
3			
4			
5	G6899	16	
6			

二者合并为一张工作表并预处理为上图格式

始发站和终到站更换为进场车站和离场车站

车站信息

基于地图大小略作修改去掉了京广铁路正线,仅保留联络线部分



单个车站及进场离场布置

库函数部分及常规列车信息部分

- 导入库函数，处理表格以及替换字符并导出
- 使用字典建立常规信息的映射关系

In [ ]:

```
import numpy
import pandas
import datetime
import random

Excelpath = "汉口站晚间.xlsx"  # Excel时刻表文件路径
TextPath = "train.txt"  # 游戏时刻表文件路径
# 速度和编组以及类型映射关系,0为普速1为动车2为高速
species = {'K': ['120', 'LPPPPPP', 0], 'T': ['140', 'LPPPPPP', 0], 'Z': ['160', 'LPPPPPP', 0],
           'D': ['200', 'LPPL', 1], 'C': ['200', 'LPPL', 1], 'G': ['300', 'LPPLLPPL', 2]}


# 车站-编号,掉向,用时以及运行车辆种类映射关系
# 图片左(0)右(1)侧线路key值相同则掉向,
# 国铁车辆行走左侧,2为数据为左侧股道编号
# [车站编号,车站所在侧(0为左侧),车辆进场股道,车辆离场行走股道,到达中心车站所用时间,行走车辆类型]
station = {'汉口动车所': ['a', 1, 0, 0, 4, 'GDC'], '京广铁路联络线丹水池方向(外侧)': ['b', 1, 2,
           '汉孝城际孝感东方向': ['d', 1, 2, 1, 4, 'GD'], '汉口站': ['e', -1, 0, 0, 0, 'KTZCDG']
           '沪蓉线重庆方向(中间)': ['g', 0, 1, 2, 4, 'D'], '汉丹线丹江口方向(最内侧)': ['h', 0, 1
ThisStation = '汉口站'
TotalPlat = 18


# track = {}
# 车型关系--待筛选，默认短编
marshalling = {}

# 始发终到映射关系--待筛选
dst = {}

# 股道到达,图片左右侧线路,,key值不同则掉向
# turnst = {'武九客专武汉方向': 0, '武九客专鄂州方向': 1, '武冈城际黄冈方向': 1}

trainDF = pandas.DataFrame(
    columns=['车次', '类型', '最高时速', '动拖布置', 'flag'])  # 列车整体信息
arriveStDF = pandas.DataFrame(columns=['车站名称', '股道', '到达时间', '停站时间'])  # 进场信息
stopStDF = pandas.DataFrame(columns=['车站名称', '股道', '到达时间', '停站时间'])  # 停站信息
leaveStDF = pandas.DataFrame(columns=['车站名称', '股道', '到达时间', '停站时间'])  # 离场信息

# 存储字符串形式的最终结果
trainList = []
arriveStList = []
stopStList = []
leaveStList = []
```

读取表格及数据处理部分

- 1,读取表格并去除空行
- 2,从车次提取速度等级
- 3,筛选始发站终到站建立替换字典

车次字符串生成部分

1,车次信息部分

- 车次号,类型,动拖布置及flag保持默认
- 速度等级依照KTZDG等区分映射

2,停站部分

- 停站2(主要车站) 到时及停站时间来自表格部分,股道由由车站大屏生成,终到车则为任意股道
- 进场及离场部分 依照游戏先期测试进行平移推算,股道依照国铁车辆靠左行驶安排,

In [ ]:

```
# 读取文件
sheet0 = pandas.read_excel(io=Excelpath, sheet_name="Sheet0")  # 车次及始发终到信息
sheet1 = pandas.read_excel(io=Excelpath, sheet_name="Sheet1")  # 检票口信息

# 去除空行
#sheet0 = sheet0.dropna()
sheet0 = sheet0[~(sheet0["车次"].isnull())]  # 删掉空行
sheet1 = sheet1.dropna()  # 去除空值

sheet = pandas.merge(sheet0, sheet1, how="left", on="车次")
sheet = sheet.reset_index(drop=True)  # 按照车次信息进行连接，合并检票口sheet1
sheet = sheet.drop_duplicates(subset="车次")
# sheet.replace("--:--",numpy.nan())
# sheet=sheet.fillna(value=0)

print(sheet)

trainInfo = sheet["车次"].to_frame()  # 车辆信息
# 统计始发终到车站信息
ts = pandas.concat([sheet0["始发站"].value_counts(),
                    sheet0["终到站"].value_counts()])
totalStation = ts.index

print(set(list(totalStation)))  # 所有始发站和终到站统计
```

```
      车次        到时        开时   始发站   终到站   检票口
0      Z4  23:09:00  23:33:00  重庆北  北京西  13.0
1     Z96  23:15:00  23:40:00  重庆西  北京西  11.0
2   D5810  19:45:00  20:00:00  宜昌东   汉口   NaN
3   D3011  19:56:00  20:26:00   上海   汉口   NaN
4   D2278  19:57:00  20:27:00  重庆北   汉口   NaN
5   G6784  20:18:00  20:48:00   巴东   汉口   NaN
6   D5986  20:19:00  20:49:00  宜昌东   汉口   NaN
7   D3033  20:37:00  21:07:00   上海   汉口   NaN
8   D5964  20:40:00  21:10:00  宜昌东   汉口   NaN
9    D630  20:55:00  21:25:00  重庆北   汉口   NaN
10  G6838  20:56:00  21:26:00  十堰东   汉口   NaN
11  D7056  21:24:00  21:54:00  云梦东   汉口   NaN
12  D3015  21:33:00  22:03:00   上海   汉口   NaN
13  D3261  21:43:00  22:13:00   福州   汉口   NaN
14  G3472  21:45:00  22:15:00  昆明南   汉口   NaN
15  C5028  21:55:00  22:25:00  咸宁南   汉口   NaN
16   D634  21:55:00  22:25:00  成都东   汉口   NaN
17  D5994  22:05:00  22:35:00   恩施   汉口   NaN
18  G6788  22:05:00  22:35:00   巴东   汉口   NaN
19  D3157  22:07:00  22:37:00   南通   汉口   NaN
20  D5812  22:15:00  22:45:00  宜昌东   汉口   NaN
21  D2881  22:25:00  22:55:00  连云港   汉口   NaN
22  C5512  22:47:00  23:17:00  大冶北   汉口   NaN
23  G3458  22:55:00  23:10:00  重庆北   汉口   NaN
24  D3047  23:03:00  23:23:00   上海   汉口   NaN
25  D5990  23:21:00  23:31:00  宜昌东   汉口   NaN
26  D5974  23:26:00  23:46:00  宜昌东   汉口   NaN
27  G3468  23:29:00  23:49:00  成都东   汉口   NaN
28  D5860  23:44:00  23:55:00  希水南   汉口   NaN
29  D5966  23:48:00  23:58:00  宜昌东   汉口   NaN
30  D5784  20:45:00  21:07:00   利川  黄冈东  13.0
31  D5757  20:08:00  20:22:00  云梦东  黄冈西  14.0
32  D5828  20:29:00  20:38:00  宜昌东   武汉   3.0
33  D5820  21:18:00  21:22:00  宜昌东   武汉   3.0
34  D5824  22:10:00  22:23:00  宜昌东   武汉   3.0
35  G6899  19:10:00  19:40:00   汉口  襄阳东  16.0
36  G6891  20:19:00  20:49:00   汉口  襄阳东  17.0
37  G1769  20:20:00  20:40:00   上海  襄阳东  14.0
38  G6897  20:50:00  21:20:00   汉口  襄阳东  17.0
39  G1527  21:38:00  21:58:00  北京西  襄阳东  12.0
40  G6895  21:47:00  22:17:00   汉口  襄阳东  15.0
41  D5811  20:00:00  20:30:00   汉口  宜昌东   NaN
42  G1515  20:01:00  20:05:00  北京西  宜昌东   3.0
43  D2189  20:15:00  20:23:00  杭州西  宜昌东   2.0
44  G1033  20:28:00  20:35:00  深圳北  宜昌东   2.0
45  D5965  20:30:00  21:00:00   汉口  宜昌东   NaN
46  G1037  20:38:00  20:47:00  深圳北  宜昌东   2.0
47  D2177  21:19:00  21:27:00  杭州西  宜昌东   2.0
48  D5241  19:40:00  19:46:00   仙桃  云梦东  17.0
{'杭州西', '咸宁南', '昆明南', '十堰东', '恩施', '连云港', '宜昌东', '云梦东', '福州', '成都东', '大冶北', '利川', '重庆北', '巴东', '黄冈东', '汉口', '希水南', '武汉', '深圳北', '黄冈西', '上海', '北京西', '仙桃', '襄阳东', '南通', '重庆西'}
```

始发终到替换

存在多路径问题

- 如从汉口站到重庆北站，动车经过沪蓉线，高速动车则经由汉孝城际,郑渝高铁至重庆北站。目前不知道怎么解决
- 手动替换准确度低

In [ ]:

```python
# 手动建立映射关系,别的方法不会


def getArrLeaSt(type, station):
    speedType=type[0]
    stcode = ""
    # 多路径车站部分
    if speedType == "G":  # 高速
        ArrLeaveSt = {
            '武汉': '沪蓉线南京南方向(内侧)',
            '重庆西': '汉孝城际孝感东方向',
            '重庆北': '汉孝城际孝感东方向',
            '成都东': '汉孝城际孝感东方向'
        }
    elif speedType == "D" or speedType == "C":  # 动车城际
        ArrLeaveSt = {
            '武汉': '京广铁路联络线丹水池方向(外侧)',
            '重庆西': '沪蓉线重庆方向(中间)',
            '重庆北': '沪蓉线重庆方向(中间)',
            '成都东': '沪蓉线重庆方向(中间)',
        }
    else:  # 普速
        ArrLeaveSt = {
            '重庆北': '汉丹线丹江口方向(最内侧)',
            '重庆西': '汉丹线丹江口方向(最内侧)',
        }

    stcode = ArrLeaveSt.get(station)
    if stcode != None:
        print("车次:{0} 车站:{1} ".format(type, station))
        return stcode

    # 常规单路径车站
    ArrLeaveSt = {
        '北京西': '沪蓉线南京南方向(内侧)',
        '济南西': '沪蓉线南京南方向(内侧)',
        '青岛': '沪蓉线南京南方向(内侧)',
        '南京南': '沪蓉线南京南方向(内侧)',
        '嘉兴南': '沪蓉线南京南方向(内侧)',
        '襄阳东': '汉孝城际孝感东方向',
        '福州': '京广铁路联络线丹水池方向(外侧)',
        '利川': '沪蓉线重庆方向(中间)',
        '昆明南': '汉孝城际孝感东方向',
        '希水南': '京广铁路联络线丹水池方向(外侧)',
        '深圳北': '京广铁路联络线丹水池方向(外侧)',
        '大冶北': '京广铁路联络线丹水池方向(外侧)',
        '连云港': '沪蓉线南京南方向(内侧)',
        '成都东': '汉孝城际孝感东方向',
        '汉口': '汉口动车所',
        '温州南': '京广铁路联络线丹水池方向(外侧)',
        '厦门': '京广铁路联络线丹水池方向(外侧)',
        '南通': '沪蓉线南京南方向(内侧)',
        '仙桃': '沪蓉线重庆方向(中间)',
        '上海': '沪蓉线南京南方向(内侧)',
        '宁波': '沪蓉线南京南方向(内侧)',
        '广州南': '沪蓉线南京南方向(内侧)',
        '郑州东': '沪蓉线南京南方向(内侧)',
        '杭州西': '京广铁路联络线丹水池方向(外侧)',
        '巴东': '汉孝城际孝感东方向',
        '十堰东': '汉孝城际孝感东方向',
        '黄冈西': '京广铁路联络线丹水池方向(外侧)',
        '黄冈东': '京广铁路联络线丹水池方向(外侧)',
        '南昌': '京广铁路联络线丹水池方向(外侧)',
        '宜昌东': '沪蓉线重庆方向(中间)',
        '咸宁南': '汉口汉西联络线汉西站方向(最外侧)',
```

```
        '恩施': '沪蓉线重庆方向(中间)',
        '利川': '沪蓉线重庆方向(中间)',
        '云梦东': '汉孝城际孝感东方向'}

    stcode = ArrLeaveSt.get(station)
    if stcode != None:
        return stcode
    else:
        print("车次:{0} 车站:{1} 未找到".format(type, station))
        exit()
```

In [ ]:
```
# 生成车次信息
# '列车编号','类型','最高时速','动拖布置','flag' 列车整体信息

for index, row in trainInfo.iterrows():
    trainNum = row["车次"]
    maxspeed = (species.get(trainNum[0]))[0]  # 有字头的列车
    if maxspeed == None:  # 最高速度未找到
        maxspeed = '120'  # 普客

    marshalling = (species.get(trainNum[0]))[1]
    # trainTC=marshalling.get() #动拖布置
    trainDF.loc[index] = [trainNum, "COMMUTER", maxspeed, marshalling, "X1"]
    trainList.append("{0} {1} {2} {3} {4} : ".format(
        trainNum, "COMMUTER", maxspeed, marshalling, "X1"))
trainDF.head()
```

Out[ ]:

	车次	类型	最高时速	动拖布置	flag
0	Z4	COMMUTER	160	LPPPPPP	X1
1	Z96	COMMUTER	160	LPPPPPP	X1
2	D5810	COMMUTER	200	LPPL	X1
3	D3011	COMMUTER	200	LPPL	X1
4	D2278	COMMUTER	200	LPPL	X1

In [ ]:
```
# 生成停站信息
for index, row in sheet.iterrows():
    arriveSt = getArrLeaSt(row["车次"], row["始发站"])
    leaveSt = getArrLeaSt(row["车次"], row["终到站"])

    # 上下行编号区分车辆进场股道为1,2道,进场及离场股道
    # 靠左行走
    arriveTrack = station.get(arriveSt)[2]
    leaveTrack = station.get(leaveSt)[3]

    # 停站股道,依照检票口选择,没有则填为0，任意股道
    # 使用int()去掉float类型的.0,防止因为格式问题读不出股道
    row = row.fillna(0)
    stopTrack = int(row["检票口"])

    at = row["到时"]

    lt = row["开时"]

    # 统一时间格式

    strTime1 = datetime.datetime.strptime(str(at), "%H:%M:%S")
    strTime2 = datetime.datetime.strptime(str(lt), "%H:%M:%S")

    stopTime = (strTime2-strTime1).seconds/60  # 分钟为单位的停站时间

    # 时间处理部分,由到达时刻推出进场时刻
    arrTime1 = strTime1-datetime.timedelta(minutes=(station.get(arriveSt))[4])
    arrTime2 = str(arrTime1.strftime('%Y-%m-%d %H:%M:%S'))[-8:]  # 仅保留时分秒
    arrTime = datetime.datetime.strptime(arrTime2, "%H:%M:%S")  # 转换格式

    # 离场时刻
    leaveTime1 = strTime2+datetime.timedelta(minutes=(station.get(leaveSt))[4])
    leaveTime2 = str(leaveTime1.strftime('%Y-%m-%d %H:%M:%S'))[-8:]
    leaveTime = datetime.datetime.strptime(leaveTime2, "%H:%M:%S")

    # 进场
    arriveStDF.loc[index] = [arriveSt, arriveTrack, arrTime2, 0]
    arriveStList.append("{0}#{1}#{2}#{3}".format(
        (station.get(arriveSt))[0], arriveTrack, arrTime2, 0))
    # 停站
    stopStDF.loc[index] = [ThisStation, stopTrack, row["到时"], stopTime]
    stopStList.append("{0}#{1}#{2}#{3}".format(
        (station.get(ThisStation))[0], stopTrack, row["到时"], int(stopTime)))

    # 离场
    leaveStDF.loc[index] = [leaveSt, leaveTrack, leaveTime2, 0]
    leaveStList.append("{0}#{1}#{2}#{3}".format(
        (station.get(leaveSt))[0], leaveTrack, leaveTime2, 0))
```

```
车次:Z4 车站:重庆北
车次:Z96 车站:重庆西
车次:D2278 车站:重庆北
车次:D630 车站:重庆北
车次:D634 车站:成都东
车次:G3458 车站:重庆北
车次:G3468 车站:成都东
车次:D5828 车站:武汉
车次:D5820 车站:武汉
车次:D5824 车站:武汉
```

In [ ]:
```
print(arriveStDF.head(), '\n', stopStDF.head(), '\n', leaveStDF.head())  # 部分信息
```

```
              车站名称   股道       到达时间   停站时间
0  汉丹线丹江口方向(最内侧)    1  23:05:00     0
1  汉丹线丹江口方向(最内侧)    1  23:11:00     0
2    沪蓉线重庆方向(中间)    1  19:41:00     0
3   沪蓉线南京南方向(内侧)    2  19:51:00     0
4    沪蓉线重庆方向(中间)    1  19:53:00     0
   车站名称   股道       到达时间   停站时间
0  汉口站   13  23:09:00  24.0
1  汉口站   11  23:15:00  25.0
2  汉口站    0  19:45:00  15.0
3  汉口站    0  19:56:00  30.0
4  汉口站    0  19:57:00  30.0
              车站名称   股道       到达时间   停站时间
0  沪蓉线南京南方向(内侧)    1  23:38:00     0
1  沪蓉线南京南方向(内侧)    1  23:45:00     0
2          汉口动车所    0  20:04:00     0
3          汉口动车所    0  20:30:00     0
4          汉口动车所    0  20:31:00     0
```

最终车次结果

合并车辆信息和停站信息

导出部分

导出为train.txt手动附加原文件头部之后替换原时刻表文件

In [ ]:

```
timeTable = open(file=TextPath, mode="w")  # 覆盖写入txt
finalRes = []
for i in range(0, len(arriveStList)):
    tstr = "{0}{1} {2} {3} ".format(
        trainList[i], arriveStList[i], stopStList[i], leaveStList[i])
    finalRes.append(tstr)  # 用于展示的列表
    tstr1 = tstr+'\n'
    timeTable.writelines(tstr1)  # 按行写入

timeTable.close()
finalRes
```

Out[ ]:
```
['Z4 COMMUTER 160 LPPPPPP X1 : h#1#23:05:00#0 e#13#23:09:00#24 c#1#23:38:00#0 ',
 'Z96 COMMUTER 160 LPPPPPP X1 : h#1#23:11:00#0 e#11#23:15:00#25 c#1#23:45:00#0 ',
 'D5810 COMMUTER 200 LPPL X1 : g#1#19:41:00#0 e#0#19:45:00#15 a#0#20:04:00#0 ',
 'D3011 COMMUTER 200 LPPL X1 : c#2#19:51:00#0 e#0#19:56:00#30 a#0#20:30:00#0 ',
 'D2278 COMMUTER 200 LPPL X1 : g#1#19:53:00#0 e#0#19:57:00#30 a#0#20:31:00#0 ',
 'G6784 COMMUTER 300 LPPLLPPL X1 : d#2#20:14:00#0 e#0#20:18:00#30 a#0#20:52:00#0 ',
 'D5986 COMMUTER 200 LPPL X1 : g#1#20:15:00#0 e#0#20:19:00#30 a#0#20:53:00#0 ',
 'D3033 COMMUTER 200 LPPL X1 : c#2#20:32:00#0 e#0#20:37:00#30 a#0#21:11:00#0 ',
 'D5964 COMMUTER 200 LPPL X1 : g#1#20:36:00#0 e#0#20:40:00#30 a#0#21:14:00#0 ',
 'D630 COMMUTER 200 LPPL X1 : g#1#20:51:00#0 e#0#20:55:00#30 a#0#21:29:00#0 ',
 'G6838 COMMUTER 300 LPPLLPPL X1 : d#2#20:52:00#0 e#0#20:56:00#30 a#0#21:30:00#0 ',
 'D7056 COMMUTER 200 LPPL X1 : d#2#21:20:00#0 e#0#21:24:00#30 a#0#21:58:00#0 ',
 'D3015 COMMUTER 200 LPPL X1 : c#2#21:28:00#0 e#0#21:33:00#30 a#0#22:07:00#0 ',
 'D3261 COMMUTER 200 LPPL X1 : b#2#21:38:00#0 e#0#21:43:00#30 a#0#22:17:00#0 ',
 'G3472 COMMUTER 300 LPPLLPPL X1 : d#2#21:41:00#0 e#0#21:45:00#30 a#0#22:19:00#0 ',
 'C5028 COMMUTER 200 LPPL X1 : f#1#21:51:00#0 e#0#21:55:00#30 a#0#22:29:00#0 ',
 'D634 COMMUTER 200 LPPL X1 : g#1#21:51:00#0 e#0#21:55:00#30 a#0#22:29:00#0 ',
 'D5994 COMMUTER 200 LPPL X1 : g#1#22:01:00#0 e#0#22:05:00#30 a#0#22:39:00#0 ',
 'G6788 COMMUTER 300 LPPLLPPL X1 : d#2#22:01:00#0 e#0#22:05:00#30 a#0#22:39:00#0 ',
 'D3157 COMMUTER 200 LPPL X1 : c#2#22:02:00#0 e#0#22:07:00#30 a#0#22:41:00#0 ',
 'D5812 COMMUTER 200 LPPL X1 : g#1#22:11:00#0 e#0#22:15:00#30 a#0#22:49:00#0 ',
 'D2881 COMMUTER 200 LPPL X1 : c#2#22:20:00#0 e#0#22:25:00#30 a#0#22:59:00#0 ',
 'C5512 COMMUTER 200 LPPL X1 : b#2#22:42:00#0 e#0#22:47:00#30 a#0#23:21:00#0 ',
 'G3458 COMMUTER 300 LPPLLPPL X1 : d#2#22:51:00#0 e#0#22:55:00#15 a#0#23:14:00#0 ',
 'D3047 COMMUTER 200 LPPL X1 : c#2#22:58:00#0 e#0#23:03:00#20 a#0#23:27:00#0 ',
 'D5990 COMMUTER 200 LPPL X1 : g#1#23:17:00#0 e#0#23:21:00#10 a#0#23:35:00#0 ',
 'D5974 COMMUTER 200 LPPL X1 : g#1#23:22:00#0 e#0#23:26:00#20 a#0#23:50:00#0 ',
 'G3468 COMMUTER 300 LPPLLPPL X1 : d#2#23:25:00#0 e#0#23:29:00#20 a#0#23:53:00#0 ',
 'D5860 COMMUTER 200 LPPL X1 : b#2#23:39:00#0 e#0#23:44:00#11 a#0#23:59:00#0 ',
 'D5966 COMMUTER 200 LPPL X1 : g#1#23:44:00#0 e#0#23:48:00#10 a#0#00:02:00#0 ',
 'D5784 COMMUTER 200 LPPL X1 : g#1#20:41:00#0 e#13#20:45:00#22 b#1#21:12:00#0 ',
 'D5757 COMMUTER 200 LPPL X1 : d#2#20:04:00#0 e#14#20:08:00#14 b#1#20:27:00#0 ',
 'D5828 COMMUTER 200 LPPL X1 : g#1#20:25:00#0 e#3#20:29:00#9 b#1#20:43:00#0 ',
 'D5820 COMMUTER 200 LPPL X1 : g#1#21:14:00#0 e#3#21:18:00#4 b#1#21:27:00#0 ',
 'D5824 COMMUTER 200 LPPL X1 : g#1#22:06:00#0 e#3#22:10:00#13 b#1#22:28:00#0 ',
 'G6899 COMMUTER 300 LPPLLPPL X1 : a#0#19:06:00#0 e#16#19:10:00#30 d#1#19:44:00#0 ',
 'G6891 COMMUTER 300 LPPLLPPL X1 : a#0#20:15:00#0 e#17#20:19:00#30 d#1#20:53:00#0 ',
 'G1769 COMMUTER 300 LPPLLPPL X1 : c#2#20:15:00#0 e#14#20:20:00#20 d#1#20:44:00#0 ',
 'G6897 COMMUTER 300 LPPLLPPL X1 : a#0#20:46:00#0 e#17#20:50:00#30 d#1#21:24:00#0 ',
 'G1527 COMMUTER 300 LPPLLPPL X1 : c#2#21:33:00#0 e#12#21:38:00#20 d#1#22:02:00#0 ',
 'G6895 COMMUTER 300 LPPLLPPL X1 : a#0#21:43:00#0 e#15#21:47:00#30 d#1#22:21:00#0 ',
 'D5811 COMMUTER 200 LPPL X1 : a#0#19:56:00#0 e#0#20:00:00#30 g#2#20:34:00#0 ',
 'G1515 COMMUTER 300 LPPLLPPL X1 : c#2#19:56:00#0 e#3#20:01:00#4 g#2#20:09:00#0 ',
 'D2189 COMMUTER 200 LPPL X1 : b#2#20:10:00#0 e#2#20:15:00#8 g#2#20:27:00#0 ',
 'G1033 COMMUTER 300 LPPLLPPL X1 : b#2#20:23:00#0 e#2#20:28:00#7 g#2#20:39:00#0 ',
 'D5965 COMMUTER 200 LPPL X1 : a#0#20:26:00#0 e#0#20:30:00#30 g#2#21:04:00#0 ',
 'G1037 COMMUTER 300 LPPLLPPL X1 : b#2#20:33:00#0 e#2#20:38:00#9 g#2#20:51:00#0 ',
 'D2177 COMMUTER 200 LPPL X1 : b#2#21:14:00#0 e#2#21:19:00#8 g#2#21:31:00#0 ',
 'D5241 COMMUTER 200 LPPL X1 : g#1#19:36:00#0 e#17#19:40:00#6 d#1#19:50:00#0 ']
```

In [ ]: