stations

For reference, ignored during import

StationReference Format:

Name	MayBeInitial	Platforms
a = 武九客专武汉方向	1	2, 1
b = 葛店南	1	1, 6, 4, 2, 3, 5
c = 武冈城际黄冈方向	1	1, 2
d = 武九客专鄂州方向	1	1, 2

timetable

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|
|--|--|--|--|--|--|--|--| |D3223 |COMMUTER |200 |LPPL |X1: | a#1#17:44:00#0 |b#3#17:46:00#2 |c#1#17:50:00#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

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

从路路通截图导出为Excel进行处理

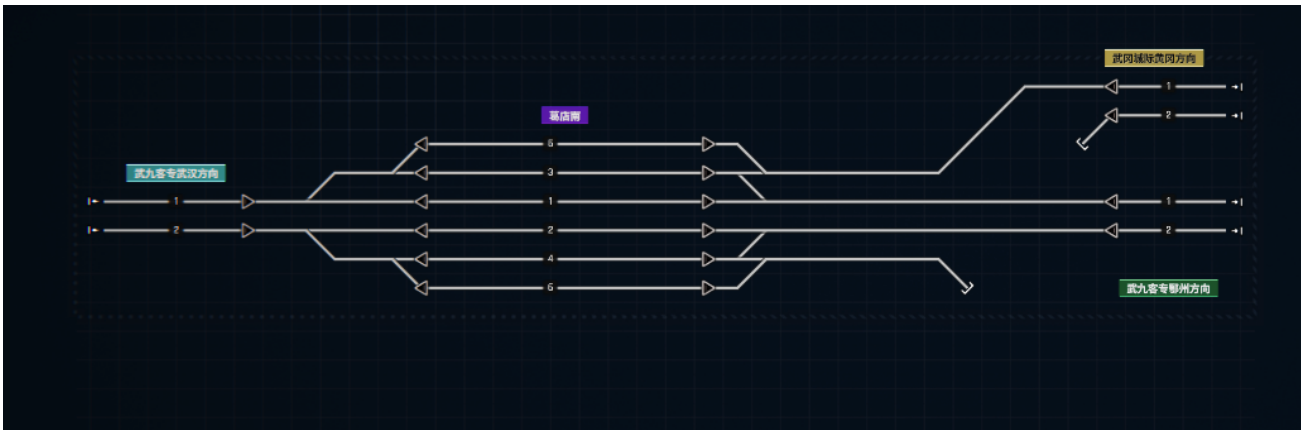
剪贴板 字体

C17 13:17:00

	A	B	C	D	E	F
16	D5765	13:10	13:12	仙桃	阳新	
17	D5853	13:15	13:17	武汉东	武穴北	
18	D3273	13:23	13:25	汉口	厦门北	
19	C5626	13:29	13:31	黄冈东	武汉	

始发站和终到站更换为进场车站和离场车站

车站信息



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

path="gdn.xlsx" #文件路径
speed = {'K': '120', 'T': '140', 'Z': '160',
         'D': '200', 'C': '200', 'G': '300'}  # 速度映射关系
# 车站-编号,股道,股道到达及用时映射关系
# 股道到达映射关系.可到达股道,图片左右侧线路key值不同则掉向,
station = {'武九客专武汉方向': ['a', '135', '0',5], '葛店南': ['b', ['123456', '',0]],
           '武冈城际黄冈方向': ['c', '46', '1',5], '武九客专鄂州方向': ['d', '24', '1',5]}
ThisStation = '葛店南'
# track = {'武九客专武汉方向': '135',
#          '武冈城际黄冈方向': '46', '武九客专鄂州方向': '24'}
# 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=['列车编号', '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 股道随机
> > 进场及离场部分 依照游戏先期测试进行平移推算 股道依照上下行安排

In [ ]:
```
# 读取文件
sheet = pandas.read_excel(io=path)

sheet = sheet.dropna()
sheet=sheet.reset_index(drop=True)

print(sheet.head())

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

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

```
   车次名称      到时      开时   始发站   终到站
0  C5602  07:23:00  07:25:00  黄冈西    武汉
1  D5770  07:53:00  07:55:00  大冶北   云梦东
2  D5782  08:23:00  08:25:00  黄冈东    利川
3  D5762  08:57:00  08:59:00  黄冈东    仙桃
4  D2181  10:42:00  10:44:00   武汉   黄梅东
{'大冶北', '杭州西', '郑州东', '利川', '仙桃', '阳新', '重庆北', '武穴北', '厦门', '黄冈西', '厦门北', '南昌西', '黄梅东', '深圳北', '梅州西', '武汉', '西安北', '武汉东', '咸宁南', '汉口', '南昌', '云梦东', '黄冈东', '宜昌东'}
```

In [ ]:
```
# 手动建立映射关系
ArrLeaveSt = {'云梦东': '武九客专武汉方向',
              '仙桃': '武九客专武汉方向',
              '利川': '武九客专武汉方向',
              '南昌': '武九客专鄂州方向',
              '南昌西': '武九客专鄂州方向',
              '厦门': '武九客专鄂州方向',
              '厦门北': '武九客专鄂州方向',
              '咸宁南': '武九客专武汉方向',
              '大冶北': '武九客专鄂州方向',
              '宜昌东': '武九客专武汉方向',
              '杭州西': '武九客专鄂州方向',
              '梅州西': '武九客专鄂州方向',
              '武汉': '武九客专武汉方向',
              '武汉东': '武九客专武汉方向',
              '武穴北': '武冈城际黄冈方向',
              '汉口': '武九客专武汉方向',
              '深圳北': '武九客专鄂州方向',
              '西安北': '武九客专武汉方向',
              '郑州东': '武九客专武汉方向',
              '重庆北': '武九客专武汉方向',
              '阳新': '武九客专鄂州方向',
              '黄冈东': '武冈城际黄冈方向',
              '黄冈西': '武冈城际黄冈方向',
              '黄梅东': '武冈城际黄冈方向'}
ArrLeaveSt
```

Out[ ]:
```
{'云梦东': '武九客专武汉方向',
 '仙桃': '武九客专武汉方向',
 '利川': '武九客专武汉方向',
 '南昌': '武九客专鄂州方向',
 '南昌西': '武九客专鄂州方向',
 '厦门': '武九客专鄂州方向',
 '厦门北': '武九客专鄂州方向',
 '咸宁南': '武九客专武汉方向',
 '大冶北': '武九客专鄂州方向',
 '宜昌东': '武九客专武汉方向',
 '杭州西': '武九客专鄂州方向',
 '梅州西': '武九客专鄂州方向',
 '武汉': '武九客专武汉方向',
 '武汉东': '武九客专武汉方向',
 '武穴北': '武冈城际黄冈方向',
 '汉口': '武九客专武汉方向',
 '深圳北': '武九客专鄂州方向',
 '西安北': '武九客专武汉方向',
 '郑州东': '武九客专武汉方向',
 '重庆北': '武九客专武汉方向',
 '阳新': '武九客专鄂州方向',
 '黄冈东': '武冈城际黄冈方向',
 '黄冈西': '武冈城际黄冈方向',
 '黄梅东': '武冈城际黄冈方向'}
```

In [ ]:
```
#'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["车次名称"]
    maxspeed=speed.get(trainNum[0])#有字头的列车
    if maxspeed==None: #最高速度未找到
        maxspeed='120' #普客

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

Out[ ]:

	列车编号	类型	最高时速	动拖布置	flag
0	C5602	COMMUTER	200	LPPL	X1
1	D5770	COMMUTER	200	LPPL	X1
2	D5782	COMMUTER	200	LPPL	X1
3	D5762	COMMUTER	200	LPPL	X1
4	D2181	COMMUTER	200	LPPL	X1

In [ ]:
```
# arriveStList=pandas.DataFrame(columns=['车站名称','股道','到达时间','停站时间'])#进场信息
# stopiveStList=pandas.DataFrame(columns=['车站名称','股道','到达时间','停站时间'])#停站信息
# leaveStList=pandas.DataFrame(columns=['车站名称','股道','到达时间','停站时间'])#离场信息
# stationInfo = sheet[["到时", "开时","始发站", "终到站"]]  # 时间及停站信息

for index, row in sheet.iterrows():
    arriveSt = ArrLeaveSt.get(row["始发站"])
    leaveSt = ArrLeaveSt.get(row["终到站"])

    # 上下行编号区分车辆进场股道为1,2道

    if int(row["车次名称"].strip()[-1]) % 2 == 0:
        arriveTrack = 2  # 偶数位上行车
    else:
        arriveTrack = 1  # 奇数下行
    # 依照掉向区分离场股道
    if (station.get(arriveSt))[2] == (station.get(leaveSt))[2]:
        leaveTrack = arriveTrack
    else:
        leaveTrack = 3-arriveTrack  # 2->1,1->2

    # 停站股道,依照映射随机选择
    stopTrack = random.choice((station.get(arriveSt)[1]))

    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

    arrTime1=strTime1-datetime.timedelta(minutes=(station.get(arriveSt))[3])
    #print((str(arrTime1))[10,-1])
    arrTime2=str(arrTime1.strftime('%Y-%m-%d %H:%M:%S'))[-8:]
    arrTime=datetime.datetime.strptime(arrTime2, "%H:%M:%S")

    leaveTime1=strTime2+datetime.timedelta(minutes=(station.get(leaveSt))[3])
    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))
```

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

```
        车站名称   股道       到达时间   停站时间
0   武冈城际黄冈方向    2  07:18:00     0
1   武九客专鄂州方向    2  07:48:00     0
2   武冈城际黄冈方向    2  08:18:00     0
3   武冈城际黄冈方向    2  08:52:00     0
4   武九客专武汉方向    1  10:37:00     0
    车站名称  股道       到达时间   停站时间
0   葛店南   4  07:23:00   2.0
1   葛店南   2  07:53:00   2.0
2   葛店南   6  08:23:00   2.0
3   葛店南   6  08:57:00   2.0
4   葛店南   1  10:42:00   2.0
        车站名称   股道       到达时间   停站时间
0   武九客专武汉方向    1  07:30:00     0
1   武九客专武汉方向    1  08:00:00     0
2   武九客专武汉方向    1  08:30:00     0
3   武九客专武汉方向    1  09:04:00     0
4   武冈城际黄冈方向    2  10:49:00     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("trains.txt",mode="a")
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)
    tstr1=tstr+'\n'
    timeTable.writelines(tstr1)

timeTable.close()
finalRes
```

Out[ ]:
```
['C5602 COMMUTER 200 LPPL X1 : c#2#07:18:00#0 b#4#07:23:00#2 a#1#07:30:00#0 ',
 'D5770 COMMUTER 200 LPPL X1 : d#2#07:48:00#0 b#2#07:53:00#2 a#1#08:00:00#0 ',
 'D5782 COMMUTER 200 LPPL X1 : c#2#08:18:00#0 b#6#08:23:00#2 a#1#08:30:00#0 ',
 'D5762 COMMUTER 200 LPPL X1 : c#2#08:52:00#0 b#6#08:57:00#2 a#1#09:04:00#0 ',
 'D2181 COMMUTER 200 LPPL X1 : a#1#10:37:00#0 b#1#10:42:00#2 c#2#10:49:00#0 ',
 'D3287 COMMUTER 200 LPPL X1 : a#1#11:01:00#0 b#5#11:06:00#2 d#2#11:13:00#0 ',
 'D5742 COMMUTER 200 LPPL X1 : c#2#11:25:00#0 b#4#11:30:00#2 a#1#11:37:00#0 ',
 'D5852 COMMUTER 200 LPPL X1 : d#2#12:01:00#0 b#4#12:06:00#2 a#1#12:13:00#0 ',
 'G2387 COMMUTER 300 LPPL X1 : a#1#12:52:00#0 b#1#12:57:00#6 d#2#13:08:00#0 ',
 'D5765 COMMUTER 200 LPPL X1 : a#1#13:05:00#0 b#1#13:10:00#2 d#2#13:17:00#0 ',
 'D5853 COMMUTER 200 LPPL X1 : a#1#13:10:00#0 b#3#13:15:00#2 c#2#13:22:00#0 ',
 'D3273 COMMUTER 200 LPPL X1 : a#1#13:18:00#0 b#3#13:23:00#2 d#2#13:30:00#0 ',
 'C5626 COMMUTER 200 LPPL X1 : c#2#13:24:00#0 b#6#13:29:00#2 a#1#13:36:00#0 ',
 'D5881 COMMUTER 200 LPPL X1 : a#1#13:36:00#0 b#5#13:41:00#2 d#2#13:48:00#0 ',
 'G2388 COMMUTER 300 LPPL X1 : d#2#13:38:00#0 b#2#13:43:00#2 a#1#13:50:00#0 ',
 'D2182 COMMUTER 200 LPPL X1 : c#2#13:49:00#0 b#6#13:54:00#3 a#1#14:02:00#0 ',
 'G2046 COMMUTER 300 LPPL X1 : d#2#14:42:00#0 b#2#14:47:00#2 a#1#14:54:00#0 ',
 'G2294 COMMUTER 300 LPPL X1 : d#2#15:01:00#0 b#2#15:06:00#3 a#1#15:14:00#0 ',
 'G2712 COMMUTER 300 LPPL X1 : d#2#15:43:00#0 b#2#15:48:00#2 a#1#15:55:00#0 ',
 'D3274 COMMUTER 200 LPPL X1 : d#2#15:51:00#0 b#2#15:56:00#2 a#1#16:03:00#0 ',
 'D5855 COMMUTER 200 LPPL X1 : a#1#16:30:00#0 b#3#16:35:00#2 c#2#16:42:00#0 ',
 'D3251 COMMUTER 200 LPPL X1 : a#1#17:17:00#0 b#1#17:22:00#2 d#2#17:29:00#0 ',
 'D5856 COMMUTER 200 LPPL X1 : c#2#17:17:00#0 b#4#17:22:00#2 a#1#17:29:00#0 ',
 'D3223 COMMUTER 200 LPPL X1 : a#1#17:39:00#0 b#3#17:44:00#2 d#2#17:51:00#0 ']
```

In [ ]:
```
```