一、基本逻辑

指定一个chronic，设定环境seed，按照donothing运行，出现gameover时根据故障录波数据找到干预时间点和干预线路，从干预线路的邻域拓扑动作（所有可重连线路、邻域变电站母线动作）以及发电机出力调整（当前述拓扑动作均无效时激活）遍历，记录能使系统运行过故障点的动作，遍历结束后，选取保证系统越过故障点后以donothing稳定运行更久的最优动作驱使系统自介入点继续运行，再次故障后重复上述操作直至该chronic运行完毕。



二、操作流程

首先运行busSet.py生成各变电站的母线动作矩阵，之后运行actiontest.py进行动作遍历

三、疑问

**1、在选取特定场景时，有何对应的指令或操作，如想要固定在一个场景上做研究，每次env.reset后是同一个场景，该如何编码（当前只能把chronics文件夹只留下待遍历的场景文件做到这一点）；正常情况下，可否选择交互场景的出现顺序（如随机抽样下一个场景，而非按照文件夹里的顺序）；**

**2、环境的seed定义尚不清楚，目前只在环境make之后，进行了env.seed(seed =1)，不确定是否实现了环境中各变量变化规律的可重复性**

3、donothing运行时是否需要加入断线重连作为强制性的自动动作，以防状态易于恶化？

**4、评判动作是否有效时哪种标准会比较好？该动作使得系统延续的时间还是reward？**

**5、是否要采取蒙特卡洛树搜索等方法进行动作的遍历，当前方法存在贪婪搜索（即每次都选择最佳的动作成为实际部署在场景中的动作）的缺点。**