# 不推荐

## Folder: PLL

The 2013 PLL code (with the acceleration of Lemma 3.16 in 2019 paper);

For both weighted and unweighted graphs;

## Folder: PLL\_with\_non\_adj\_reduction

The 2013 PLL (weighted and unweighted) code (with the acceleration of Lemma 3.16 in 2019 paper) + 2019 R1 + 2019 R2 enhance （先按2019规则选MG，再在这个基础上按不相邻规则把degree小的点加进MG里面）;

注意：用了2019 R2 enhance之后的图就是weighted，不能使用Unweighted bfs了！

## Folder: PSL\_weighted

The 2019 PSL weighted code (no reduction);

With codes of removing redundant labels (only necessary for weighted graphs, this addresses issues raised by 2019 authors);

For weighted graphs;

(最好把并行代码里面Lemma 3.16的w<u改成w<=u)

## Folder: PSL\_with\_non\_adj\_reduction

The 2019 PSL (weighted) code + 2019 R1 + 2019 R2 enhance （先按2019规则选MG，再在这个基础上按不相邻规则把degree小的点加进MG里面）; 这个代码里面的2019 R2是去掉MG里面的点再加边，不是2019文章里面R2用于PSL的方式。

## Folder: PSL\_enhancedoriginalR2

The 2019 PSL (weighted) code + 2019 R1 + 2019 R2 enhance （先按2019规则选MG，再在这个基础上按不相邻规则把degree小的点加进MG里面）; 这个代码里面的2019 R2是不去掉MG里面的点不加边（2019文章里面R2用于PSL的方式）。

## CT\_v1

基于杨同学写的CT代码，里面的MDE或许可以优化下

# Folder: two\_hop\_v1（推荐）

## PLL\_v1

基于Folder: PLL\_with\_non\_adj\_reduction，但是把所有label和R1 R2信息都整合进了case info class里面

## PSL\_v1

基于Folder: PSL\_enhancedoriginalR2。这个代码里面的2019 R2是不去掉MG里面的点不加边（2019文章里面R2用于PSL的方式）。

## CT\_v2

基于杨同学写的CT代码，优化了里面的MDE