软件工程实验一项目架构

一 需要完成的功能以及初步设计

1.1 功能需求

1. 读入文本并生成有向图
   1. 需要注意边权值：A→B的权重w=文本中A和B相邻出现的次数， w>=1
2. 展示有向图
   1. 无UI：对于每条边A→B，在控制台上输出” A -> B”，每条边占一行
   2. 有UI：使用GraphStream绘图并展示
3. 查询桥接词
4. 根据桥接词生成新文本
5. 计算两个单词之间的最短路径
   1. 无UI：对于最短路”A->B->C->D”，输出”A->B”、”B->C”、”C->D”，每个输出占一行
   2. 有UI：将最短路的边用不同的颜色突出显示
   3. 可选功能：只输入一个单词并计算到任意单词的最短路
      1. 无UI：对于每个单词，按照5.a的格式输出，不同单词之间用”\*\*\*\*\*\*\*\*\*\*”作为分割
      2. 有UI：到不同单词的最短路用不同的颜色进行区分
6. 随机游走
   1. 用户应能随时停止

1.2 初步设计

1.2.1 有向图的存储

采用邻接表的方式进行存储。同时，为每个单词分配一个index，设置单词到index和index到单词的两个映射，邻接表的节点表示用index。

1.2.2 文本到有向图

需要注意，换行、标点符号当作空格，非字母要忽略。Gpt给出的参考代码如下：



1.2.3 有向图的展示

* 无UI：遍历邻接表，查找对应index的单词，并成对输出。
* 有UI：使用GraphStream生成有向图，并展示。

1.2.4 桥接词查询

采用邻接表存储，所以当查询单词A和单词B的桥接词时

1. 查询A和B对应的index，IA和IB
2. 根据IA查找A的后继IA\_next
3. 对每个IA\_next，查询其后继是否有IB
4. 如果有则将IA\_next对应的单词加入结果集合中
5. 如果没有则返回2直到A的所有后继都被遍历完
6. 返回结果集合

1.2.5 根据桥接词生成新文本

1. 首先将用户输入的Str划分为单词列表，{word\_1,word\_2…word\_n}
2. 对于每个word\_i，根据查询桥接词的程序查找word\_i和word\_i+1的桥接词
3. 如果有则在word\_i和word\_i+1之间插入桥接词
4. 否则继续遍历
5. 将结果单词列表拼接成完整的字符串Str并返回

1.2.6 计算两个单词最短路

直接深搜，记录最短路index，在返回时将index转换为对应的单词返回。

选做：对每个目标单词调两个单词最短路，将结果合并为一个列表返回，列表中每个目标单词最短路的顺序按照其index从小到大排序。

1.2.7 随机游走

TODO：

二 基本类设计

2.1 GraphIO

作为图的输入输出接口，完成从文本构建图以及将各个功能的输出接口

2.1.1 属性

TODO：

2.1.2 Public方法

TODO