codesim实验报告

盛朱恒 MF20330066

本实验的代码相似度检测算法以课程网站上提供的示例MOSS为基础，并作出了微小的改进。

1、对代码进行文本查重

使用《Winnowing: Local Algorithms for Document Fingerprinting》中的算法。

1)将代码文件进行读取，并使用n-gram的思想，采用滑动窗口，得到子串

2)对每个窗口中的字符串计算哈希值

3)采用winnowing方法选取哈希值，即：设定N长度的窗口，扫描哈希值序列，只保留每个窗口中的最小值。

4)对比两个代码文件的哈希值，计算重复率1。计算方法类似IoU。

$$重复率 = \frac{重复哈希值数量}{代码文件1的哈希值数量 + 代码文件1的哈希值数量 - 重复哈希值数量}$$

其它：在处理文本时，将"\n","\t"处理为空格，并保证连续只有1个空格。

2、对汇编代码进行查重

考虑到如果只对文本代码进行查重，如果学生更改变量名，那么查重准确率会大幅降低，因此加入对汇编代码的查重。

使用“g++ -S”生成汇编指令文件，使用**1**中的方法进行查重。修改的地方在于，选取文本的窗口滑动时，以行为单位，而非以字符为单位。

最终得到一个重复率2。

3、得出结果

取重复率1和重复率2的平均值作为最终结果。

4、分析有效性

1、对文本的查重可以有效地查出简单的复制粘贴少量修改的抄袭行为。

2、对汇编代码的查重可以有效地查出简单的复制粘贴并批量修改变量名称的抄袭行为。同时，一些对代码结构的改变（if-else/switch），也会由于编译器的优化，导致其修改无效。

3、但是，对汇编代码查重也有明显弱点，即通过修改数据类型，导致汇编代码明显不同。例如，将int改为long long int, double等。因此，仍然保留了对文本的查重作为补充。