In [3]:

```
pip install -r requirements.txt
```

```
Collecting scikit_learn==0.24.2
  Downloading scikit_learn-0.24.2-cp39-cp39-win_amd64.whl (6.9 MB)
     ---------------------------------------- 6.9/6.9 MB 43.6 kB/s eta 0:00:0
0
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\admin\anacond
a3\lib\site-packages (from scikit_learn==0.24.2->-r requirements.txt (line
1)) (2.2.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\admin\anaconda3\lib
\site-packages (from scikit_learn==0.24.2->-r requirements.txt (line 1)) (1.
9.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\admin\anaconda3\lib
\site-packages (from scikit_learn==0.24.2->-r requirements.txt (line 1)) (1.2
3.5)
Requirement already satisfied: joblib>=0.11 in c:\users\admin\anaconda3\lib\s
ite-packages (from scikit_learn==0.24.2->-r requirements.txt (line 1)) (1.1.
0)
Installing collected packages: scikit_learn
  Attempting uninstall: scikit_learn
    Found existing installation: scikit-learn 1.0.2
    Uninstalling scikit-learn-1.0.2:
      Successfully uninstalled scikit-learn-1.0.2
Successfully installed scikit_learn-0.24.2
Note: you may need to restart the kernel to use updated packages.
```

In [6]:

```python
import os
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

student_files = [doc for doc in os.listdir() if doc.endswith('.txt')]
student_notes = [open(_file, encoding='utf-8').read()
                 for _file in student_files]


def vectorize(Text): return TfidfVectorizer().fit_transform(Text).toarray(
def similarity(doc1, doc2): return cosine_similarity([doc1, doc2])


vectors = vectorize(student_notes)
s_vectors = list(zip(student_files, vectors))
plagiarism_results = set()


def check_plagiarism():
    global s_vectors
    for student_a, text_vector_a in s_vectors:
        new_vectors = s_vectors.copy()
        current_index = new_vectors.index((student_a, text_vector_a))
        del new_vectors[current_index]
        for student_b, text_vector_b in new_vectors:
            sim_score = similarity(text_vector_a, text_vector_b)[0][1]
            student_pair = sorted((student_a, student_b))
            score = (student_pair[0], student_pair[1], sim_score)
            plagiarism_results.add(score)
    return plagiarism_results


for data in check_plagiarism():
    print(data)
```

```
('The adventure of lost key.txt', 'The mystery of hidden treasure.txt', 0.590
2291382892241)
('The Enigmatic doorway.txt', 'The mystery of hidden treasure.txt', 0.5875610
787187364)
('The Enigmatic doorway.txt', 'The adventure of lost key.txt', 0.554870534636
173)
```