In [1]:

```
# Some import statements

import math
from collections import Counter
```

Sklearn Implementation

In [4]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
vectorizer.fit(corpus)
skl_output = vectorizer.transform(corpus)

print(vectorizer.get_feature_names())
print(vectorizer.idf_)
print(skl_output.toarray())
```

```
['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third', 'thi
s']
[1.91629073 1.22314355 1.51082562 1.         1.91629073 1.91629073
 1.         1.91629073 1.        ]
[[0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]
 [0.         0.6876236  0.         0.28108867 0.         0.53864762
  0.28108867 0.         0.28108867]
 [0.51184851 0.         0.         0.26710379 0.51184851 0.
  0.26710379 0.51184851 0.26710379]
 [0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]]
```

Custom Implementation

In [6]:

```
from collections import Counter
from tqdm import tqdm
from scipy.sparse import csr_matrix
import math
import operator
from sklearn.preprocessing import normalize
import numpy as np

def CountDocumentWithTerm(Term):
    count = 0
    for document in corpus:
        for word in document.split(' '):
            if word == Term:
                count+=1
                break
    return count

def IDF(words):
    IDF_Dict = {}
    N = len(corpus)+1
    for word in words:
        sample = CountDocumentWithTerm(word)+1
        IDF_Dict[word] = 1+math.log(N/sample)
    return IDF_Dict


# fit function
def fit():
    UniqueWords = {}
    AllWordsList = []
    for document in corpus:
        AllWordsList+=document.split(' ')
    UniqueWords = dict(Counter(AllWordsList))
    return UniqueWords

def calc_TF(word, document):
    count=0
    for eachWord in document:
        if word == eachWord:
            count+=1
    return (count / len(document))


# transform function
def transform(words, IDFs):
    TFIDF_dict = []
    TFIDF_list = []
    document = corpus[0]
    for document in corpus:
        TFIDF_list = []
        for word in words:
            TF = calc_TF(word, document.split(' '))
            IDF = IDFs[word]
            TFIDF_list.append(TF*IDF)
        TFIDF_dict.append(TFIDF_list)
    return normalize(sorted(TFIDF_dict))

if __name__ == "__main__":
    UniqueWords = fit()
```

```
    IDFs = dict(IDF(sorted(UniqueWords.keys())))
    print ("----------------------FEATURES----------------------")
    print (sorted(UniqueWords.keys()))
    print ("\n\n********************OUTPUT COMPARISON*****************")
    print ("\n\n----------------------SKLEARN IDF VALUES----------------------")
    print(vectorizer.idf_)
    print ("\n\n----------------------CALCULATED IDF VALUES----------------------")
    print (list(IDFs.values()))
    print ("\n\n----------------------SKLEARN TF-IDF VALUES----------------------")
    print(skl_output.toarray())
    print ("\n\n----------------------CALCULATED TF-IDF VALUES----------------------")
    normalized_TFIDF_dict = transform(sorted(UniqueWords.keys()), IDFs)
    print (normalized_TFIDF_dict)
```

```
----------------------FEATURES----------------------
['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third', 'thi
s']


********************OUTPUT COMPARISON*****************


----------------------SKLEARN IDF VALUES----------------------
[1.91629073 1.22314355 1.51082562 1.         1.91629073 1.91629073
 1.         1.91629073 1.        ]


----------------------CALCULATED IDF VALUES----------------------
[1.916290731874155, 1.2231435513142097, 1.5108256237659907, 1.0, 1.9162907
31874155, 1.916290731874155, 1.0, 1.916290731874155, 1.0]


----------------------SKLEARN TF-IDF VALUES----------------------
[[0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]
 [0.         0.6876236  0.         0.28108867 0.         0.53864762
  0.28108867 0.         0.28108867]
 [0.51184851 0.         0.         0.26710379 0.51184851 0.
  0.26710379 0.51184851 0.26710379]
 [0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]]


----------------------CALCULATED TF-IDF VALUES----------------------
[[0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]
 [0.         0.46979139 0.58028582 0.38408524 0.         0.
  0.38408524 0.         0.38408524]
 [0.         0.6876236  0.         0.28108867 0.         0.53864762
  0.28108867 0.         0.28108867]
 [0.51184851 0.         0.         0.26710379 0.51184851 0.
  0.26710379 0.51184851 0.26710379]]
```