In [1]:

```
from konlpy.corpus import kobill

def getLexicon():
    lexicon = list()
    for document in [kobill.open(idx).read() for idx in kobill.fileids()]:
        for term in document.split():
            if term not in lexicon:
                lexicon.append(term)
    return lexicon
```

In [2]:

```
%timeit getLexicon()
```

```
109 ms ± 513 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [3]:

```
def getLexiconBySet():
    lexicon = list()
    for document in [kobill.open(idx).read() for idx in kobill.fileids()]:
        for term in document.split():
            lexicon.append(term)
    return list(set(lexicon))

lexicon = getLexiconBySet()
```

In [4]:

```
%timeit getLexiconBySet()
```

```
25.2 ms ± 432 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [5]:

```
def documentRepresentationByDict():
    documentList = list()
    for document in [kobill.open(idx).read() for idx in kobill.fileids()]:
        bow = dict()
        for term in document.split():
            bow[lexicon.index(term)] = 1
        documentList.append(bow)
    return documentList
```

In [6]:

```
%timeit documentRepresentationByDict()
```

```
207 ms ± 2 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

In [7]:

```
def documentRepresentationByDict2():
    documentList = list()
    for document in [kobill.open(idx).read() for idx in kobill.fileids()]:
        bow = dict()
        for term in document.split():
            bow[term] = 1
        documentList.append(bow)
    return documentList
```

In [8]:

```
%timeit documentRepresentationByDict2()
```

```
24.8 ms ± 243 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [10]:

```
from collections import defaultdict

def documentRepresentationByDefaultDict():
    documentList = list()
    for document in [kobill.open(idx).read() for idx in kobill.fileids()]:
        bow = defaultdict(int)
        for term in document.split():
            bow[term] += 1
        documentList.append(bow)
    return documentList
```

In [11]:

```
%timeit documentRepresentationByDefaultDict
```

```
17.8 ns ± 0.107 ns per loop (mean ± std. dev. of 7 runs, 100000000 loops each)
```

In [12]:

```
def documentRepresentationByDefaultDict2():
    documentList = defaultdict(lambda:defaultdict(int))
    for idx in kobill.fileids():
        for term in kobill.open(idx).read().split():
            documentList[idx][term] += 1
    return documentList
```

In [13]:

```
%timeit documentRepresentationByDefaultDict2()
```

```
27 ms ± 505 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

In [14]:

```
docList = documentRepresentationByDefaultDict2()
```

In [15]:

```
query = '국회 의원'

def booleanResult():
    result = list()
    for term in query.split():
        searchResult = list()
        for idx, termList in docList.items():
            if term in termList.keys():
                searchResult.append(idx)
        result.append(searchResult)
    one = result.pop()
    while result:
        temp = result.pop()
        one = list(set(one).intersection(temp))
    return one
```

In [16]:

```
%timeit booleanResult
```

20.6 ns ± 0.16 ns per loop (mean ± std. dev. of 7 runs, 10000000 loops each)

In [17]:

```
TDM = defaultdict(lambda:defaultdict(int))
for idx, termList in docList.items():
    for term, freq in termList.items():
        TDM[term][idx] = freq

def booleanResult2():
    result = list()
    for term in query.split():
        result.append(list(TDM[term].keys()))
    one = result.pop()
    while result:
        temp = result.pop()
        one = list(set(one).intersection(temp))
    return one
```

In [18]:

```
%timeit booleanResult2
```

21.9 ns ± 0.135 ns per loop (mean ± std. dev. of 7 runs, 10000000 loops each)

In [ ]:

```

```