21. 이진탐색 (Binary Search) Algorithm

문제

색인이 붙어있지 않은 영어 사전이 정렬된 리스트로 주어져 있을 때 주어진 단어를 가장 빨리 찾아낼 수 있는 프로그램을 작성하라. 영어 사전은 임의의 text 를 이용하여 작성하고, 정규식(regular expression)을 이용하여 영어 단어 이외에는 모두 제거한다.

이진탐색 알고리즘에서 최대 탐색 횟수는 $\approx log_2(n)$ 이라고 알려져 있다. 위 문제에서 최대 탐색 횟수는 얼마인가 ?

- 알고리즘

  1. 중간 위치($\frac{n}{2}$ 번째)의 단어를 먼처 찾고 주어진 단어와 같은지 비교한다.
  2. 찾은 단어가 주어진 단어보다 크면 $\frac{n}{2}$ 보다 큰 쪽은 버리고 작은 쪽 절반(1~ $\frac{n}{2}$ 번)을 대상으로 다시 중간 위치 ($\frac{\frac{n}{2}}{2}$ 번째) 단어를 찾아서 크기를 비교하는 작업을 주어진 단어를 찾을 때까지 반복한다.

     반대로 중간위치(n/2 번째) 단어가 주어진 단어보다 작으면 큰쪽 절반 ($\frac{n}{2} + 1$ ~ n 번)을 대상으로 같은 작업을 반복한다.

In [1]:

```
1  text = "Alice's Adventures in Wonderland (commonly shortened to Alice in Wonderland) is an 1
```

In [2]:

```
1  text
```

Out[2]:

"Alice's Adventures in Wonderland (commonly shortened to Alice in Wonderland) is an 1865 novel written by English author Charles Lutwidge Dodgson under the pseudonym Lewis Carroll.[1] It tells of a young girl named Alice falling through a rabbit hole into a fantasy world populated by peculiar, anthropomorphic creatures. The tale plays with logic, giving the story lasting popularity with adults as well as with children.[2] It is considered to be one of the best examples of the literary nonsense genre.[2][3] Its narrative course, structure, characters, and imagery have been enormously influential[3] in both popular culture and literature, especially in the fantasy genre."

text 중간의 [1], [2], [3] ... 제거

In [3]:

```
1  import re
2
3  text = re.sub(r'\[\d\]', '', text)
```

영문 대문자, 소문자 이외의 문자 모두 제거

In [4]:

```
1 text = re.sub(r'[^A-Za-z ]', '', text)
2 text
```

Out[4]:

'Alices Adventures in Wonderland commonly shortened to Alice in Wonderland is an novel written by English author Charles Lutwidge Dodgson under the pseudonym L ewis Carroll It tells of a young girl named Alice falling through a rabbit hole into a f antasy world populated by peculiar anthropomorphic creatures The tale plays with l ogic giving the story lasting popularity with adults as well as with children It is cons idered to be one of the best examples of the literary nonsense genre Its narrative c ourse structure characters and imagery have been enormously influential in both p opular culture and literature especially in the fantasy genre'

word index dictionary 작성

In [5]:

```
words = text.split()
words = sorted(list(set(words)))
words
```

Out[5]:

```
['Adventures',
 'Alice',
 'Alices',
 'Carroll',
 'Charles',
 'Dodgson',
 'English',
 'It',
 'Its',
 'Lewis',
 'Lutwidge',
 'The',
 'Wonderland',
 'a',
 'adults',
 'an',
 'and',
 'anthropomorphic',
 'as',
 'author',
 'be',
 'been',
 'best',
 'both',
 'by',
 'characters',
 'children',
 'commonly',
 'considered',
 'course',
 'creatures',
 'culture',
 'enormously',
 'especially',
 'examples',
 'falling',
 'fantasy',
 'genre',
 'girl',
 'giving',
 'have',
 'hole',
 'imagery',
 'in',
 'influential',
 'into',
 'is',
 'lasting',
 'literary',
 'literature',
 'logic',
 'named',
 'narrative',
```

'nonsense',
'novel',
'of',
'one',
'peculiar',
'plays',
'popular',
'popularity',
'populated',
'pseudonym',
'rabbit',
'shortened',
'story',
'structure',
'tale',
'tells',
'the',
'through',
'to',
'under',
'well',
'with',
'world',
'written',
'young']

In [6]:

```
1 len(words)
```

Out[6]:

78

In [7]:

```
1 words[39]
```

Out[7]:

'giving'

이진 탐색 함수

In [14]:

```
def findword(word, words):
    start = 0
    end  = len(words)
    search_count = 0

    while (start < end):
        search_count += 1
        middle = len(words) // 2          # 가운데의 위치

        if words[middle] == word:         # found
            return search_count
        elif words[middle] > word:        # 찾으려는 단어가 아래쪽 절반에 위치
            start, end = 0, middle
        else:
            start, end = middle+1, len(words)   # 찾으려는 단어가 위쪽 절반에 위치

        words =  words[start : end]
```

In [15]:

```
findword('creatures', words)
```

Out[15]:

7

In [17]:

```
findword('rabbit', words)
```

Out[17]:

6

In [18]:

```
findword('giving', words)
```

Out[18]:

1

최대 탐색 횟수

words 의 단어 중 가장 탐색 횟수가 많은 것

In [12]:

```
max_count = 0

for word in words:
    count = findword(word, words)
    if count > max_count:
        max_count = count
print("max_count =", max_count)
```

max_count = 7

평균 탐색 횟수

words 의 전체 단어 탐색 횟수를 평균

In [20]:

```
total_search_count = 0

for word in words:
    count = findword(word, words)
    total_search_count += count
average_count = total_search_count / len(words)
print("average_count =", average_count)
```

average_count = 5.461538461538462

연습문제

- 위와 동일한 작업을 한글 문서로 연습
- spyder 로 code 작성

In [21]:

```
lorem = """대법원장은 국회의 동의를 얻어 대통령이 임명한다. 헌법재판소는 법률에 저촉되지 아니하는

국군은 국가의 안전보장과 국토방위의 신성한 의무를 수행함을 사명으로 하며, 그 정치적 중립성은 준수된
```

In [15]:

```
lorem
```

Out[15]:

'대법원장은 국회의 동의를 얻어 대통령이 임명한다. 헌법재판소는 법률에 저촉되지 아니하는 범위안에서 심판에 관한 절차, 내부규율과 사무처리에 관한 규칙을 제정할 수 있다. 감사원은 원장을 포함한 5인 이상 11인 이하의 감사위원으로 구성한다. 의무교육은 무상으로 한다. 언론·출판에 대한 허가나 검열과 집회·결사에 대한 허가는 인정되지 아니한다.\n\n국군은 국가의 안전보장과 국토방위의 신성한 의무를 수행함을 사명으로 하며, 그 정치적 중립성은 준수된다. 헌법재판소는 법관의 자격을 가진 9인의 재판관으로 구성하며, 재판관은 대통령이 임명한다. 국무총리는 국회의 동의를 얻어 대통령이 임명한다. 모든 국민은 종교의 자유를 가진다. 국방상 또는 국민경제상 긴절한 필요로 인하여 법률이 정하는 경우를 제외하고는, 사영기업을 국유 또는 공유로 이전하거나 그 경영을 통제 또는 관리할 수 없다.'

In [16]:

```
word_list = lorem.split()
word_list.sort()
```

In [ ]:

```
def findword(word):
    start = 0
    end = len(word_list)
    words = word_list
    search_count = 0

    while(start < end):
        search_count += 1
        middle = len(words) // 2

        # CODE HERE

        words = words[start:end]

print(findword('헌법재판소는'))

max_count = 0
for word in word_list:
    count = findword(word)
    if count > max_count:
        max_count = count

print("max_count = ", max_count)

total_search_count = 0
for word in word_list:
    count = findword(word)
    total_search_count += count

average_cnt = total_search_count / len(word_list)

print('average_cnt = {:.2f}'.format(average_cnt))
```