IE260 Homework #5 Report

20150359 Chiyoon Park

1. Code Description

- PriorityLinkedList Part

```
def insertWithPriority(self,node):
    current = self.head
    while(True):
        if current.priority < node.priority:
            prevNode = current.getPrev()
            node.setNext(current)
            prevNode.setNext(node)
            node.setPrev(prevNode)
            current.setPrev(node)
            break
        current = current.getNext()
```

insertWithPriority 메소드의 경우 node의 priority를 원소들과 비교하여 priority가 tail 방향으로 갈수록 작아지는 형태로 node를 삽입하는 메소드입니다. 이를 위하여 while문을 사용하여 head부터 현재 LinkedList를 탐색합니다. 여기서 current의 priority보다 node의 priority가 작은 경우, current와 그 전 노드 사이에 node를 삽입하여야 priority 순서와 삽입 순서를 모두 고려한 위치에 node가 들어가게 됩니다. 따라서 prevNode에 현재 노드의 이전 노드를 저장한 후, prevNode와 current를 node의 양 끝으로 설정하여 node를 그 사이에 삽입하여줬습니다.

```
if __name__ == "__main__":

    list = PriorityLinkedList()
    list.insertWithPriority(PriorityNode(100, 'a'))
    list.insertWithPriority(PriorityNode(100, 'b'))
    list.insertWithPriority(PriorityNode(90, 'c'))
    list.insertWithPriority(PriorityNode(80, 'd'))
    list.insertWithPriority(PriorityNode(110, 'e'))

    print(list)
```

```
e(110) -> a(100) -> b(100) -> c(90) -> d(80) ->

Process finished with exit code 0
```

위와 같이 insertWithPriority 메소드를 완성한 후 위의 코드를 실행시킨 결과 오른쪽과 같이 나중에 삽입한 원소가 tail 방향으로 삽입되고, priority가 작은 원소가 tail 방향으로 삽입되는 결과를 얻을 수 있었습니다.

- createWordDocumentTable Part

```
def createWordDocumentTable(self):
    for itrWord in range(len(self.words)):
        strWord = self.words[itrWord]
        lstWordFrequency = PriorityLinkedList()
        for itrDoc in range(self.numDocuments):
            if itrWord in self.bows[itrDoc].keys():
                lstWordFrequency.insertWithPriority(PriorityNode(self.bows[itrDoc][itrWord], self.files[itrDoc]))
        self.hashWordDocument.put(strWord, lstWordFrequency)
```

createWordDocumentTable는 key가 단어, Value가 그 단어의 횟수에 따라 배열된 file을 데이터로 가지는 링크드리스트인 hash를 만드는 메소드입니다. BagOfWordCreator 클래스에서 생성된 Bow의 경우 Document에 어떤 word가 몇 회 사용되었는지에 대한 정보를 가지고 있습니다. 따라서 위와 같은 hash를 생성하기 위해선, word를 하나씩 탐색하면서 해당 word를 가지고 있는 Document를 확인해야 합니다. 따라서 for문을 두 개를 사용하여 밖의 for문은 사용된 word를 저장해둔 words를 전부 탐색하여 문서에 등장한 모든 단어를 탐색하게 하였습니다. 이후 안쪽 for문에선 모든 Document를 탐색하여 해당 word가 있는지 검사하게 하였습니다. 여기서 검사한 결과를 저장하기 위한 priorityLinkedList를 선언하여 이를 lstWordFrequency에 저장하였습니다. 이후 만약 해당 word가 해당 문서 안에 들어있는 경우, 이 횟수를 priority, 문서의 이름을 value로 하는 PriorityNode를 생성하여 lstWordFrequency에 insertWithPriority로 삽입하였습니다. 그리고 이 lstWordFrequency를 value로 하고, key를 해당 단어 strWord로 하여 클래스 내의 변수 hashWordDocument에 put하였습니다.

- Query Part

```
def query(self,strQuery):
    lstQueryWords = strQuery.split()
    hashResult = HashTable(100)
    for strQueryWord in lstQueryWords:
        for char in ' ?.!/;:()-<>_₩₩[]{}':
            strQueryWord = strQueryWord.replace(char,'')
        strQueryWord = strQueryWord.lower()
        lstPriorityDocs = self.hashWordDocument.get(strQueryWord)
        if lstPriorityDocs == None:
            continue
        nodePriorityDoc = lstPriorityDocs.getHead().getNext()
        while(nodePriorityDoc.getPriority() != -99999):
            intNewPriority = nodePriorityDoc.getPriority()
            if hashResult.get(nodePriorityDoc.getValue()) != None:
                intNewPriority = intNewPriority + hashResult.get(nodePriorityDoc.getValue())
            hashResult.put(nodePriorityDoc.getValue(), intNewPriority)
            nodePriorityDoc = nodePriorityDoc.getNext()
```

```
lstFrequency = []
lstDocument = []

lstKeysInResult = hashResult.keys
for key in lstKeysInResult:
    lstFrequency.append(hashResult.get(key))
    lstDocument.append(key)

sorting = SortingAlgorithm.QuickSort()
lstDocument, lstFrequency = sorting.performSorting(lstFrequency,lstDocument)

cnt = 0
for itr in range(len(lstDocument)):
    self.printItem(lstDocument[itr],lstFrequency[itr])
    cnt = cnt + 1
    if cnt > 4:
        break
```

Query는 input을 통해서 사용자에게 입력 받은 문자열을 Document를 조사하여, 그 단어가 어디에 몇 개씩 있는지 보여주는 메소드입니다. 먼저 input받은 문자열을 공백을 기준으로 split하여 단어를 나누어 lstQueryWords에 저장했습니다. 그리고 이 메소드의 실행 결과를 저장할 hashResult를 선언하였습니다. 이후 위에서 자른 문자열을 for문을 통해 하나씩 탐색합니다. 그리고 그 문자를 처리를 하여 특수문자를 제거하고, 모두 소문자로 변경합니다. 그리고 이 단어를 createWordDocumentTable 메소드를 통해 생성된 hashWordDocument에서 찾아서 그 단어의 정보를 가지고 있는 priorityLinkedList를 얻습니다. 그리고 이 리스트의 노드를 head 다음 노드부터 tail까지 하나씩 탐색합니다. 이를 위해서 while문 내부의 조건을 tail의 priority인 -9999가 아닌 동안으로 설정하였습니다. 여기서 if문을 통해서 hashResult에 이미 값이 저장되어 있는 경우, 앞에서 계산한 값과 현재 계산한 단어의 사용 횟수를 더하게 하였습니다. 그리고 이를 다시 hashResult에 저장하면서 리스트를 전부 탐색하게 하였습니다.

2. Result

```
!!!!! Doubling Hash !!!!!
!!!!! Doubling Hash !!!!!
Initialized.............................
Num of Document : 3000
Num of Word : 64506

Search Phrase : iran
25 : ./20_newsgroups/77228
 NNTP-Posting-Host: po4.andrew.cmu.edu  In-Reply-To: <39298@optima.cs.arizona.edu>       Some articles on the topic:
11 : ./20_newsgroups/75909
   There have been a number of articles on the PBS frontline program  about Iranian bomb. Here is my $0.02 on this and rel
9 : ./20_newsgroups/178415
 Date: Sun, 18 Apr 1993 19:21:51 GMT    In article <C5LIHI.389@ccu.umanitoba.ca> ebrahim@ee.umanitoba.ca (Mohamad Ebrahimi
9 : ./20_newsgroups/75932
 Date: Sun, 18 Apr 1993 19:21:51 GMT    In article <C5LIHI.389@ccu.umanitoba.ca> ebrahim@ee.umanitoba.ca (Mohamad Ebrahimi
8 : ./20_newsgroups/76144
           I would like to share with netters a few points I picked up from the PBS      Frontline program regarding Irar

Search Phrase : iran iraq
!!!!! Doubling Hash !!!!!
33 : ./20_newsgroups/77228
 NNTP-Posting-Host: po4.andrew.cmu.edu  In-Reply-To: <39298@optima.cs.arizona.edu>       Some articles on the topic:
21 : ./20_newsgroups/77401
 In-Reply-To: <1993May16.170102.9690@cs.wisc.edu>      Excerpts from netnews.talk.politics.mideast: 16-May-93 Re: Saudi  c
11 : ./20_newsgroups/75909
   There have been a number of articles on the PBS frontline program  about Iranian bomb. Here is my $0.02 on this and rel
11 : ./20_newsgroups/178415
 Date: Sun, 18 Apr 1993 19:21:51 GMT    In article <C5LIHI.389@ccu.umanitoba.ca> ebrahim@ee.umanitoba.ca (Mohamad Ebrahimi
11 : ./20_newsgroups/75932
 Date: Sun, 18 Apr 1993 19:21:51 GMT    In article <C5LIHI.389@ccu.umanitoba.ca> ebrahim@ee.umanitoba.ca (Mohamad Ebrahimi

Search Phrase : white house
!!!!! Doubling Hash !!!!!
!!!!! Doubling Hash !!!!!
!!!!! Doubling Hash !!!!!
37 : ./20_newsgroups/54684
 Nntp-Posting-Host: bwa.kgn.ibm.com  Organization: IBM Kingston NY  Lines: 1835      Try the firearms archive.  Larry Cipr
36 : ./20_newsgroups/179034
                                      THE WHITE HOUSE                        Office of the Press Secretary  __________
35 : ./20_newsgroups/179073
                                          THE WHITE HOUSE                        Office of the Press Secretary  ______
20 : ./20_newsgroups/178314
                                    THE WHITE HOUSE                        Office of the Press Secretary  ________________
13 : ./20_newsgroups/176910
 Organization: LNK Corporation, Riverdale, MD  Lines: 152  Nntp-Posting-Host: descartes.tec.army.mil    In article <1993Ap
```