

과제 3. 연결리스트

과제 3

- ◆ 제출 마감일 : 5월 20일 (월요일) 10:59pm
- ◆ eClass 과제방에 제출
- ◆ 사용 언어
 - 문제별 언어 지정
- ◆ 제출 양식
 - 보고서 (표지 + 문제 내용과 해결 방안 + 결과 캡처)
 - 소스코드 (별도 파일로 제출)
- ◆ 보고서는 PDF 중 하나로 제출

타인의 과제를 복사하지 말 것

- ◆ 제공된 사전 파일의 단어들을 연결리스트로 만든 후, 사전 프로그램을 만든다.
- ◆ 단어 사전 파일
 - randdict.txt 파일에는 약 5만개(48406개)의 단어와 단어의 뜻이 저장되어 있다. 별도의 저장 순서는 없다. 즉, 랜덤한 순서로 저장되어 있다.
 - 단어와 단어의 뜻은 “:” 으로 구분되어 있다.
- ◆ 단어 사전 데이터로 연결리스트 만들기
 - 자료형은 연결 리스트를 이용한다. 연결리스트의 종류(단순, 원형, 이중)는 알아서 선택한다. 그 연결리스트를 선택한 이유는 무엇인가?
 - 연결리스트에 저장할 때에는 알파벳 순으로 정렬된 상태로 저장한다. 따라서 파일에서 단어를 하나 가져와 정렬된 순서로 연결리스트에 단어를 넣어야 한다.

(1) 연결 리스트 - 사전 만들기

◆ 기능

- 단어 검색하기 : 단어를 입력하면 해당 단어의 뜻을 표시한다.

>> apple

n.사과

(1) 연결 리스트 - 사전 만들기

◆ 왜 연결리스트로 만들었을까?

- 5만개가 넘는 데이터가 연결리스트로 구성되어 있다.
- 사전 데이터는 추가/삭제할 일이 거의 없지만, 연결리스트의 이해를 위해 사용하였다.
- 사전에서 단어를 찾을 때에는 정렬해 두는 것이 편하니 파일에서 불러올 때 정렬된 상태로 연결리스트에 저장한다.
 - 즉, 파일은 정렬되어 있지 않지만, 데이터를 읽어 연결리스트를 만들 때 정렬된 상태로 구성한다.
 - 별도의 정렬 과정을 거치는 것은 아니다.

데이터 파일을 수정하면 안됨

(2) 정렬의 성능 평가

◆ 성능 평가하기

- 우리는 연결리스트가 아니라 배열에 사전을 저장할 수도 있다.
- 동일한 사전 파일을 (1) 배열에 저장한 후 정렬하는 것과 (2) 연결리스트로 구성하면서 자연스럽게 정렬하는 것을 비교하여 어느 것이 더 효과적인지 실제 시간을 측정하여 비교하라.
 - 예) 배열에 담은 후 정렬 : 5.231초 소요
연결리스트에 담으며 정렬 : 5.231초 소요
- 배열에 담은 내용을 정렬할 때에는 우리가 이미 배운 선택 정렬을 이용하라. 퀵정렬이나 힙정렬, 또는 환경이 제공하는 정렬 방법을 쓰면 안 된다.
 - 이것은 정렬되어 있지 않은 파일을 배열에 담은 후 정렬하는 것과 연결리스트에 넣으면서 정렬하는 것을 비교하기 위함이다.

(2)는 (1)에 포함시켜 하나의 프로그램으로 구성할 수도 있다.

◆ (1)에서 했던 작업을 Python으로 구현한다.

- 사전 데이터를 가져와 간단한 사전 프로그램을 작성한다. 파이썬에서 연결리스트로 작성한다. 연결리스트는 LinkedList 클래스를 작성하여 사용한다.
- 소스코드를 참고하는 것도 좋다. 그러나 참고 소스코드는 pos 번호로 node를 찾기 위해 get_node를 사용하고 있으나, 이것은 바람직하지는 않다. pos 번호를 쓰지 않는 연결리스트로 작성해야 한다.
- 단어 검색하기 : 단어를 입력하면 해당 단어의 뜻을 표시한다.

>> apple

n.사과

◆ 개선하기

- (3)에서 연결리스트로 구현한 사전에서 단어를 찾을 때, 이미 정렬되어 있음에도 불구하고 원하는 데이터를 찾을 때에는 순차 탐색을 할 수밖에 없다.
- 어떻게 하면 좀 더 빠르게 원하는 단어를 찾을 수 있을까? **방법을 제안하고 구현하라.**
- 어느 정도 효과적인지 비교하라.
- 연결리스트의 구조를 일부 변경할 수 있다.

◆ 예)

기존 연결리스트 저장 구조에서 랜덤한 5만개의 단어를 검색할 때 : 5.231초

개선된 연결리스트 저장 구조에서 랜덤한 5만개의 단어를 검색할 때 : 4.231초

(4)는 (3)과 결합하여 하나의 하나의 프로그램으로 구성할 수도 있다.