TRIE를 이용한 SPELLING CORRECTION 시스템 개발

Ⅰ. 목 표

□ **TRIE를 사용하여 아래의 목적을 달성 할 수 있도록 프로젝트 코드를 작성하고, 코드를 실행하여 목적을 달성하는지 실험한다.**

- 1단계: (레코드 저장) 주어진 파일의 정보를 모두 삽입하는 코드를 작성한다.
  - 해당 단계에서 작성이 필요한 함수는 다음과 같으며, 수업 중 설명내용을 참고하여 작성한다.
  - guide 파일에 제공된 함수의 틀을 이용한다.
    ▷ search_trie, hang_down, insert_trie를 완성한다.

- 2단계: (기능 개발) TRIE를 탐색하는 코드를 작성한다.
  - 해당 단계에서 작성이 필요한 함수는 다음과 같으며, 수업 중 배운 내용은 참고하여 작성한다.
    ▷ get_penalty, output_top_corrections, substitution, deletion, insertion, transposition

- 3단계: (탐색 실험) TRIE에서 탐색을 수행하는 작업을 실험한다.

Ⅱ. 수행 방법

□ **1단계: (레코드 저장) 주어진 파일의 정보를 모두 삽입하는 코드를 작성한다.**

- 파일 "Corpus_dictionalry_AP_Penn_RARE.txt"는 각 줄마다 word, windex, wfrequency가 있다.
  - 각 줄은 빈칸으로 구분되며 main함수에서 각 줄을 각각의 레코드로 만든다.
  - 이 레코드의 key는 word를 가진다.
- 각 레코드를 Trie에 저장한다.
  - key는 trie에 저장하고 레코드는 master 파일(binary file)에 저장한다.
  - Trie에서 key의 마지막 노드("₩0"을 가진 노드)는 master 파일에서 해당 레코드의 위치(byte address)를 가지도록 한다.
- 위의 과정을 수행함에 있어 필요한 코드인 search_trie, hang_down, insert_trie를 작성한다.

□ **2단계: (기능 개발) TRIE를 탐색하는 코드를 작성한다.**

- 탐색할 키(한 영어 단어)를 입력 받아 탐색을 수행하는 작업과, 그 과정에서 spell correction을 수행하는 find_with_spell_correction을 작성할 수 있도록 한다.
  - 탐색 과정에서 교정 후보를 발견하게 되는데, 발견 후보는 교정 결과 테이블 cwords에 저장하도록 한다.
- get_penalty와 output_top_corrections를 완성하여 아래를 수행 할 수 있도록 한다.
  - 만일, trie에 입력단어와 완전히 매칭되는 것이 있다면 아래와 같이 출력할 수 있도록 한다.

```
TYPE A KEY: helpless
total number of corrected results = 1

< 0>: helpless  penalty:   0.000,      substitution: 0,  delettion: 0,  insertion: 0,  transposition: 0
```

- 완벽히 매칭되는 것이 없다면 벌점(penalty)이 작은 순서대로 상위 10개 교정결과를 출력하도록 한다.
  ▷ 벌점은 다음과 같이 부여할 수 있도록 한다.
  substitution: 1.1; deletion: 1.3; insertion: 1.6; transposition: 1.9.
  ▷ 탐색 과정에서 교정은 2까지만 시도한다. 그 이상의 교정은 수행하지 않도록 한다.

<예시1>

```
TYPE A KEY: hld
total number of corrected results = 10045

< 0>: had       penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 1>: hid       penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 2>: Old       penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 3>: old       penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 4>: held      penalty:   1.300,       substitution: 0,  delettion: 1,  insertion: 0,  transposition: 0
< 5>: hold      penalty:   1.300,       substitution: 0,  delettion: 1,  insertion: 0,  transposition: 0
< 6>: hae       penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 7>: ham       penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 8>: han       penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 9>: has       penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
```

<예시3>

```
TYPE A KEY: hlod
total number of corrected results = 4422

< 0>: hood      penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 1>: clod      penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 2>: plod      penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 3>: hold      penalty:   1.900,       substitution: 0,  delettion: 0,  insertion: 0,  transposition: 1
< 4>: hand      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 5>: hard      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 6>: he'd      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 7>: head      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 8>: heed      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 9>: held      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
```

<예시2>

```
TYPE A KEY: hapy
total number of corrected results = 5428

< 0>: hazy      penalty:   1.100,       substitution: 1,  delettion: 0,  insertion: 0,  transposition: 0
< 1>: happy     penalty:   1.300,       substitution: 0,  delettion: 1,  insertion: 0,  transposition: 0
< 2>: hay       penalty:   1.600,       substitution: 0,  delettion: 0,  insertion: 1,  transposition: 0
< 3>: hace      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 4>: hack      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 5>: hail      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 6>: hair      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 7>: hajj      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 8>: half      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
< 9>: hall      penalty:   2.200,       substitution: 2,  delettion: 0,  insertion: 0,  transposition: 0
```

- 교정 과정(spell correction)을 수행하는데 필요한 코드는 다음과 같다. 아래 코드를 완성한다.
  - substitution, deletion, insertion, transposition

- $ 가 탐색할 단어이면 프로그램이 종료된다.
- 총 삽입되는 Total record는 94121개 이다.

□ **3단계 다음과 같이 TRIE에서 탐색을 수행하는 작업을 실험한다. 실험을 진행하고 실행 화면을 다음과 같이 캡쳐한다.**

- **실험 단어:** computer, compuer, happy, hpy, world, wolrd

Ⅲ. 제출물

□ **제출**

- **제출물은 다음의 파일을 압축하여 제출한다.**
  - guide를 기반으로 하여 작성된 c파일
  - 실험결과를 캡처한 이미지
- **제출물인 압축파일과 c파일, 이미지 파일은 다음과 같이 파일 이름을 수정하여 제출한다.**
  - 학번_이름.zip
  - 학번_이름.c
  - 학번_이름_실험단어.jpg (이미지 파일: jpg, jpeg, png 등)
- **예시**
  - 2022000111_홍길동.zip
  - 2022000111_홍길동.c
  - 2022000111_홍길동_computer.jpg, 2022000111_홍길동_compuer.jpg 등

- **참고**
  - 소스와 txt파일은 ANSI로 인코딩 되어 있다.