[2020카카오공채]
가사 검색

2019.11.13

최성국

수준별 접근법

모르겠다

이중포문

효율성 테스트가 있을것이다

효율적인 검색을 위한 자료구조

문제

문제 설명

[본 문제는 정확성과 효율성 테스트 각각 점수가 있는 문제입니다.]

친구들로부터 천재 프로그래머로 불리는 프로도는 음악을 하는 친구로부터 자신이 좋아하는 노래 가사에 사용된 단어들 중에 특정 키워드가 몇 개 포함되어 있는지 궁금하니 프로그램으로 개발해 달라는 제안을 받았습니다.

그 제안 사항 중, 키워드는 와일드카드 문자중 하나인 '?'가 포함된 패턴 형태의 문자열을 뜻합니다. 와일드카드 문자인 '?'는 글자 하나를 의미하며, 어떤 문자에도 매치된다고 가정합니다. 예를 들어 `"fro??"`는 `"frodo"`, `"front"`, `"frost"` 등에 매치되지만 `"frame"`, `"frozen"`에는 매치되지 않습니다.

가사에 사용된 모든 단어들이 담긴 배열 `words`와 찾고자 하는 키워드가 담긴 배열 `queries`가 주어질 때, 각 키워드 별로 매치된 단어가 몇 개인지 순서대로 배열에 담아 반환하도록 `solution` 함수를 완성해 주세요.

문제 해설

- 정답률 : 정확성: 34.4%, **효율성: 0.8%**
- 문자열을 다룰 수 있는지 파악
- 트라이 자료구조 또는 이분 탐색등 보다 효율적인 방법을 이용해 코드를 작성할 수 있는지 파악

- 2020 신입 개발자 블라인드 채용 1차 코딩 테스트 문제 해설

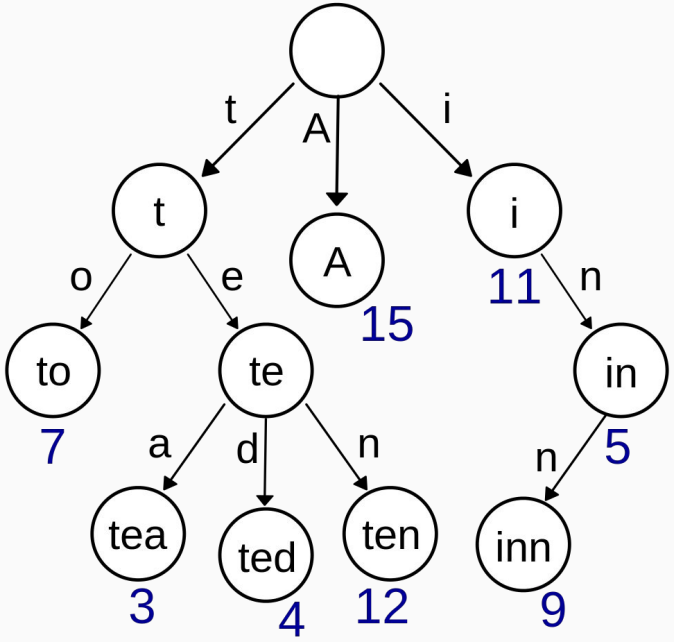
출제 의도

- 정답률 : 정확성: 34.4%, 효율성: 0.8%
- 문자열을 다룰 수 있는지 파악
- 트라이 자료구조 또는 이분 탐색등 보다 효율적인 방법을 이용해 코드를 작성할 수 있는지 파악

효율적인 검색을 위한 저장

- 2020 신입 개발자 블라인드 채용 1차 코딩 테스트 문제 해설

Trie 구조



retrieval tree

- 빠른 문자열 탐색을 위한 자료구조

+ 본 문제에서는 ???o 와 같은 쿼리 처리를 위하여 뒤집힌 word로 tree를 한개 더 만들어야 한다

자바를 통한 구현

```java
class Node {
    char name;
    ArrayList<Node> child = new ArrayList<>();
    boolean end = false;
    int numOfChildren = 0;

    public Node(char name) {
        super();
        this.name = name;
    }

    public void insertChild(char c) {
        child.add(new Node(c));
    }

    public void checkEnd() {
        this.end = true;
    }

    public Node getOneChild(char c) {
        Node ans = null;
        for (Node ch : child) {
            if (ch.name == c) {
                ans = ch;
                break;
            }
        }
        return ans;
    }
```

Node class의 멤버 변수

1. 이름
2. 자식 리스트
3. 끝 여부
4. 총 자식 수

나는
어떻게
고통받았나

트라이 구조 구현 후
모든자식노드탐색

```
테스트 1 〉 실패 (시간 초과)
테스트 2 〉 통과 (1627.52ms, 509MB)
테스트 3 〉 실패 (시간 초과)
테스트 4 〉 통과 (1497.69ms, 312MB)
테스트 5 〉 통과 (609.38ms, 353MB)
```

1차시도

노드 별 자식의
길이의 Max값 도입

```
테스트 1 〉 실패 (시간 초과)
테스트 2 〉 통과 (1646.03ms, 453MB)
테스트 3 〉 실패 (시간 초과)
테스트 4 〉 통과 (1003.89ms, 246MB)
테스트 5 〉 통과 (640.48ms, 355MB)
```

2차시도

글자수 별 root를 따로
만든다(1 ~ 100001)

```
테스트 1 〉 실패 (시간 초과)
테스트 2 〉 통과 (1323.64ms, 456MB)
테스트 3 〉 실패 (시간 초과)
테스트 4 〉 통과 (706.66ms, 289MB)
테스트 5 〉 통과 (749.03ms, 387MB)
```

3차시도

글자수 별 root를 따로
만든다(있는 길이만)

```
테스트 1 〉 실패 (시간 초과)
테스트 2 〉 통과 (1343.30ms, 426MB)
테스트 3 〉 실패 (시간 초과)
테스트 4 〉 통과 (570.35ms, 227MB)
테스트 5 〉 통과 (612.51ms, 354MB)
```

4차시도

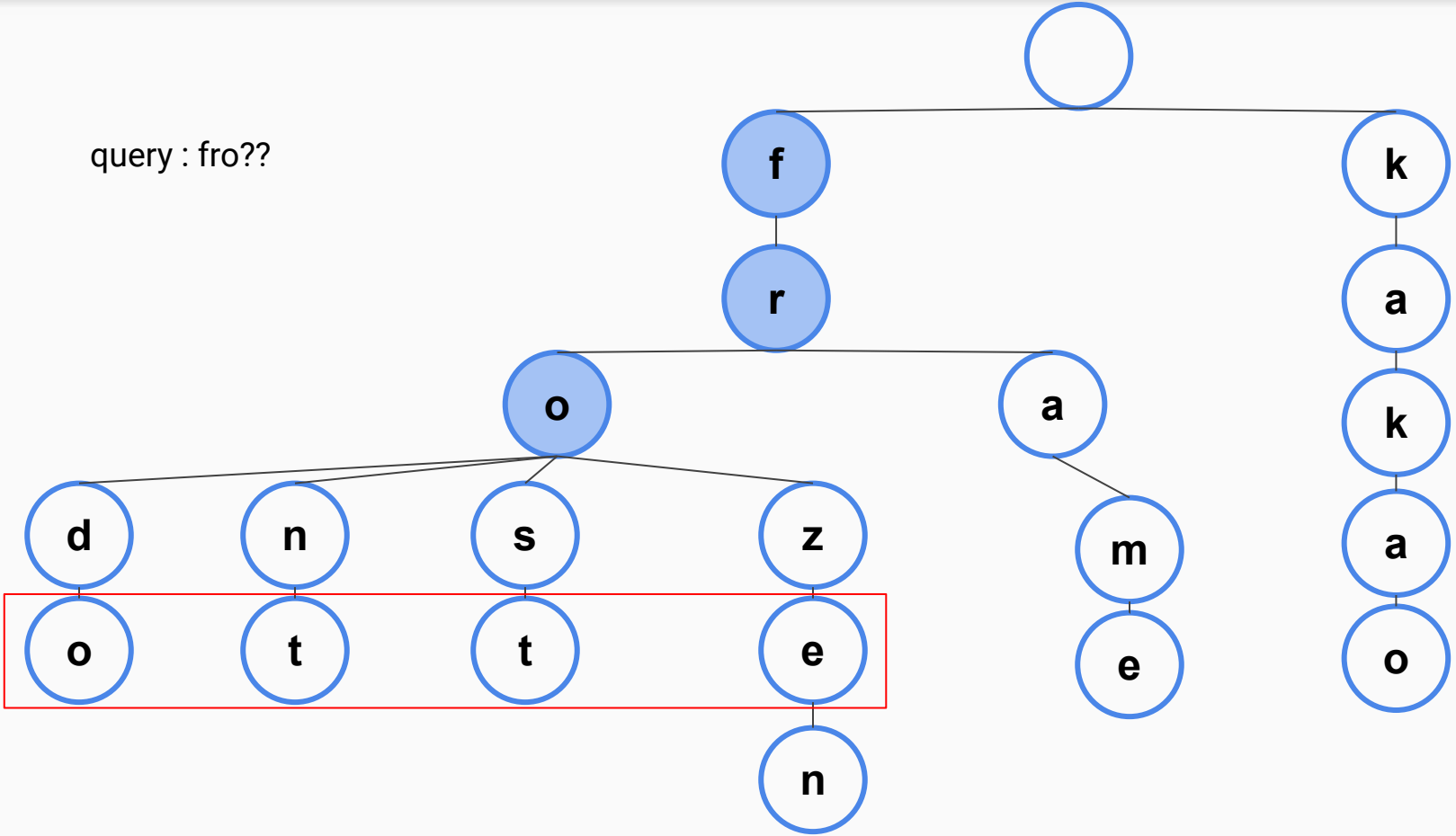
노드의 멤버 변수로
자식의 수 도입

```
테스트 1 〉 통과 (258.80ms, 114MB)
테스트 2 〉 통과 (463.98ms, 192MB)
테스트 3 〉 통과 (469.67ms, 188MB)
테스트 4 〉 통과 (342.64ms, 181MB)
테스트 5 〉 통과 (531.64ms, 318MB)
```

5차시도

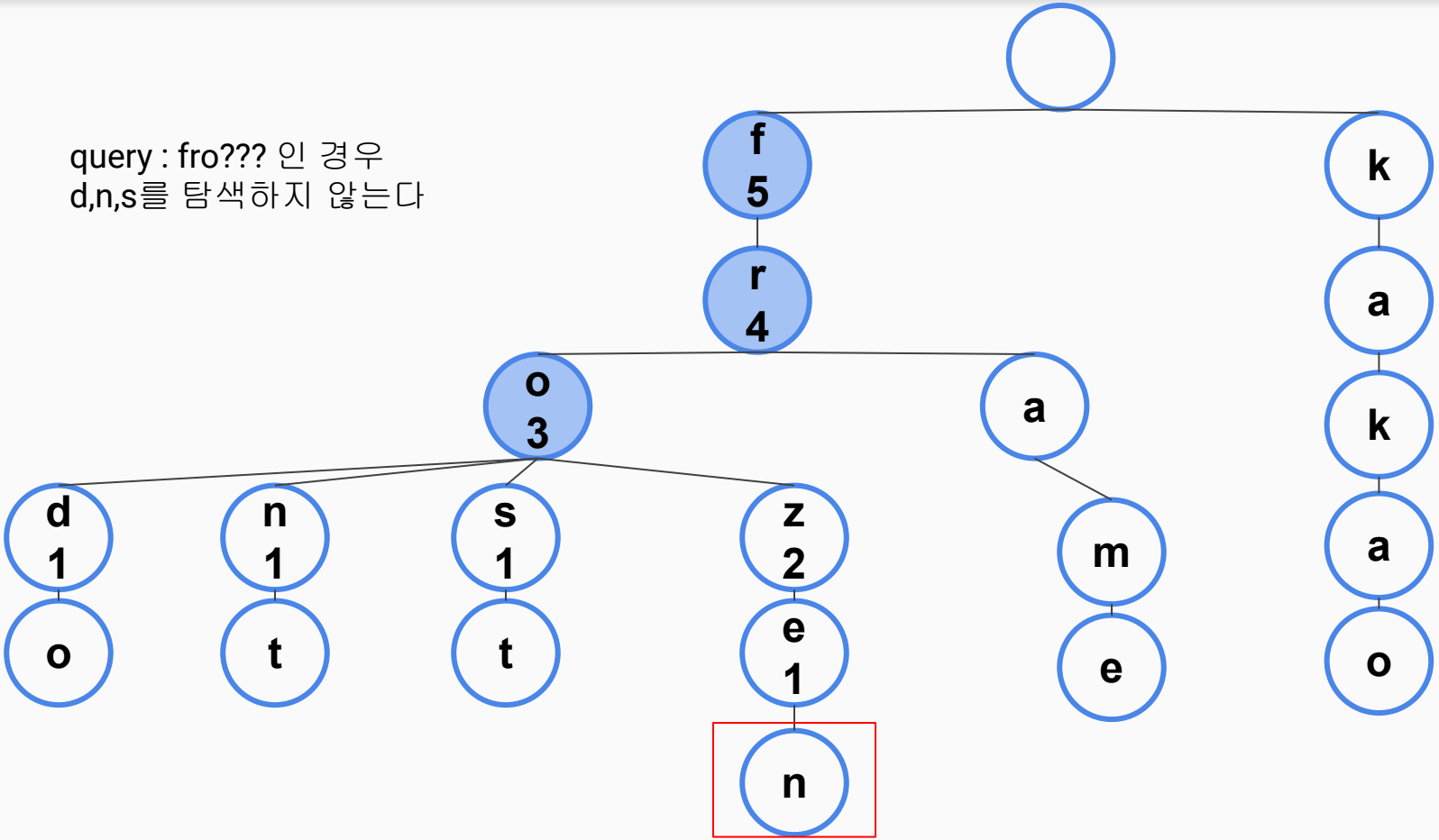
1차 시도 (Trie 구조 구현 후 모든 자식노드 탐색)

query : fro??



2차 시도 (노드별 자식 길이의 Max 할당)

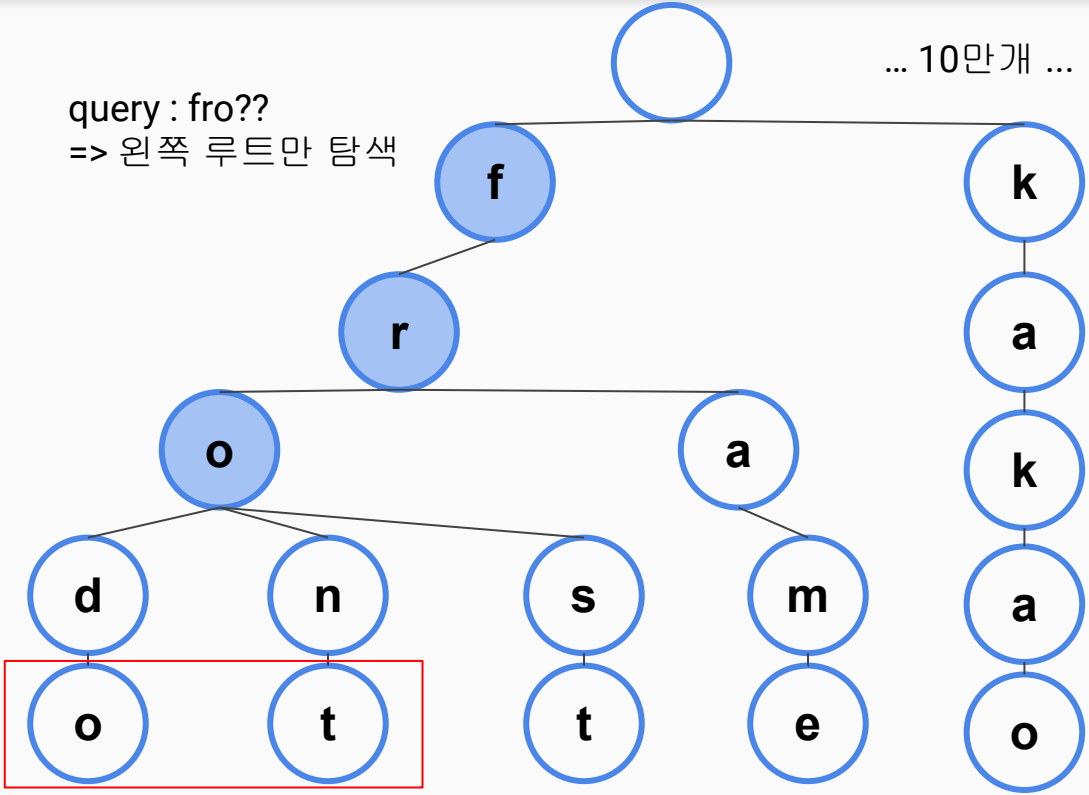
query : fro??? 인 경우
d,n,s를 탐색하지 않는다



3차 시도 (단어 길이 별로 Trie 구조를 따로 만든다)

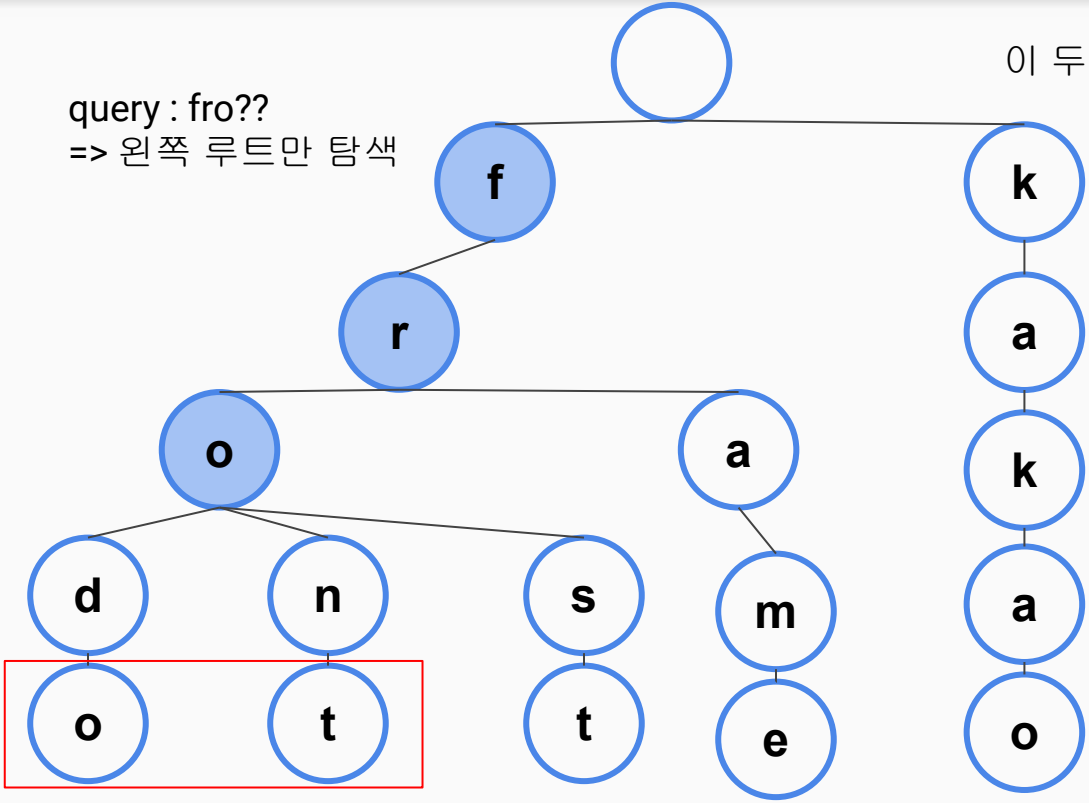
query : fro??
=> 왼쪽 루트만 탐색

... 10만개 ...



4차 시도 (단어 길이 별로 Trie 구조를 따로 만든다)

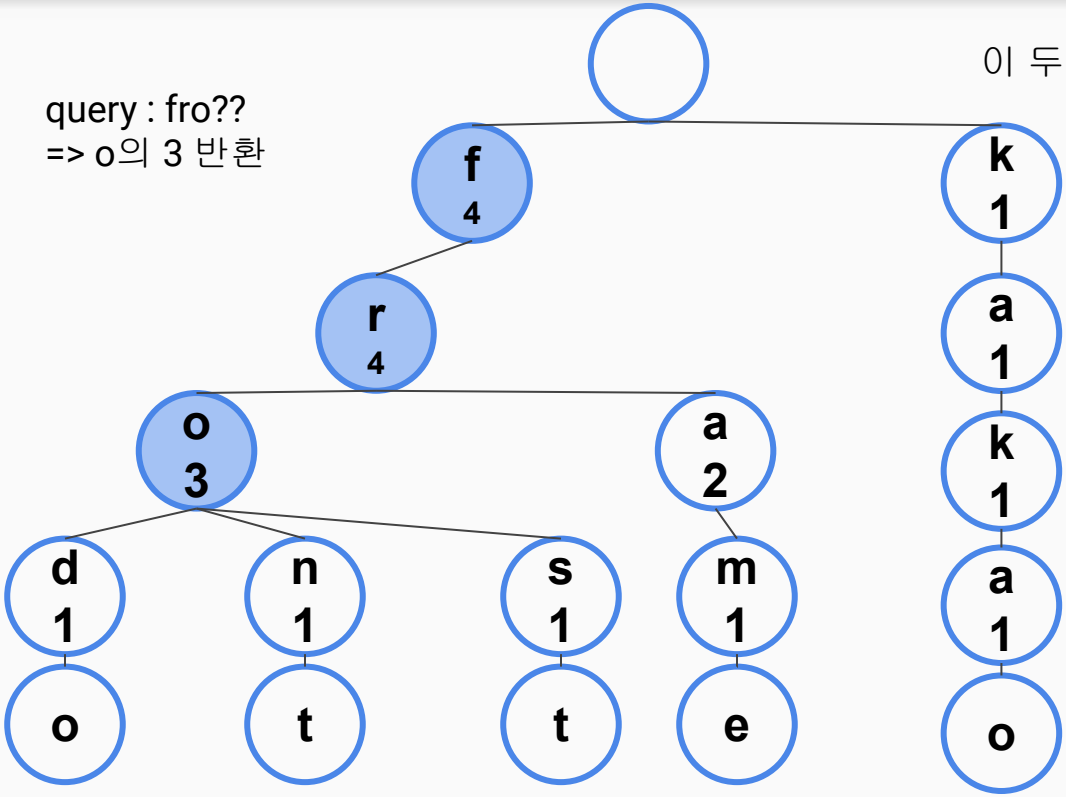
query : fro??
=> 왼쪽 루트만 탐색



이 두개만

5차 시도 (노드 별 자식 개수 변수를 할당)

query : fro??
=> o의 3 반환



이 두개만

감사합니다