Concurrent Programming Assignment1, Signal 컴퓨터공학부 성예닮

Task

주어진 시그널에서 검색 단어를 찾아서 출력한다.  
없으면 -1을 출력하고, 검색 단어와 시그널 모두 소문자 알파벳(‘a’ – ‘z’)으로만 구성된다.

Example

|  |  |
| --- | --- |
| Input | Output |
| 3  apple  pineapple  application  Q penpineappleapplepen  A app  Q penpineappleapplepen  A tropine  D apple  Q abcdefghijklmnopqrstropineappleuvwxyz | R  pineapple|apple  pineapple|app|apple  tropine|pineapple|app |

Solution

다중 패턴 매칭 알고리즘인 Aho-Corasick 알고리즘을 이용한다.  
Aho-Corasick 알고리즘은 세 가지 자료구조를 이용한다.(자세한 설명은 Data structure 파트 참고)  
1. Trie  
2. Failure link  
3. Output link

검색 단어 사전에 apple, pineapple, application, 시그널이 penpineappleapplepen일 때, apple을 찾아가는 과정을 살펴본다.

1. apple, pineapple, application에 대한 Trie를 생성한다.  
2. Trie에서 a를 탐색할 때 까지 p, e, n, …, Failure link를 통해 a까지 탐색한다.  
3. a이후 p, p, l, e 까지 탐색 후 Output link에 의해 매칭 후, 찾았다고 마킹한다.

Data structure

Trie:

K-진 트리의 형태로 마치 우리가 사전을 찾는 것과 같은 방식으로 단어를 찾는다.(현재 구현에서는 소문자 알파벳의 개수인 26진 트리이다. ‘apple’이라는 단어를 찾으려면 ‘a’부터 차례로 ‘p’, … ‘e’를 찾는다.  
이러한 자료구조는 공간을 많이 차지하는 단점이 있지만, 문자열의 길이에 비례하는 탐색 속도를 가진다.

Failure link:

문자열 탐색 중 더 이상 연결된 노드가 없다면 처음으로 돌아가지 않고, 현재 문자와 같은 문자로 연결해주는 Failure link를 이용하여 해당 노드부터 탐색한다.  
Failure link는 Trie에서 BFS를 이용하여 생성한다.

Make failure link:  
1레벨 노드들의 failure link는 Root이다.  
현재 위치(c)와, 가리키는 노드(t)에 대해서:  
c의 failure link(cF)에서 t와 같은 문자의 노드(s)가 있다면, t의 failure link(tF)가 s를 가리킨다.  
같은 문자의 노드가 존재하지 않는다면, c <- cF 후 다시 탐색한다.

Output link:

패턴 중 다른 패턴의 일부인 문자열이 존재한다면, 전체 시그널에 대한 모든 패턴들을 탐색할 수 없기 때문에, Output link를 사용한다.  
각각 패턴 문자열이 끝나는 노드는, 해당 노드가 패턴 문자열의 끝임을 알려주는 Output link가 필요하다.

Usage

project1 디렉토리에서 make 후 ./run 을 통해 실행한다.

TODO:

싱글 스레드로 작성된 코드라 멀티 쓰레드로 구현된 코드에 비해 느리게 동작함을 볼 수 있다.  
멀티 쓰레드를 이용하여 문자열을 탐색하면 더 빠르게 탐색이 가능할 것 같다.