INHA UNIVERSITY

ICE3020 알고리즘설계

<보이어 무어 알고리즘 과제>

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.
2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.
3. 나는 보고서의 내용을 조작하지 않겠습니다.
4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.
5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2022년 05 월 8 일

학부 정보통신공학
학년 3
성명 이성인
학번 12171820

1. 개요

보이어-무어 알고리즘을 아래와 같은 제약조건을 만족하도록 구현하고 동작여부를 보일 것

2. 상세 설계 내용

```
    string txt="";
    string line;
    ifstream file("txt.txt"); // txt.txt 파일을 연다. 없으면 생성.
    if(file.is_open()){
        while(getline(file, line)) {
            txt+=line;
        }
        file.close(); //  파일을 닫는다.
    } else {
        cout << "Unable to open file";
        return 1;
    }
//12171820 이성인
```

File을 읽어와서 해당. Txt에 저장합니다.

```
//12171820 이성인
    string pattern = "letter \"A\"..\"Z\"";
    BoyerMoore(txt, pattern);

return 0;
```

저장하고 pat에 찾아낼 값들을 바꿔가며 찾습니다.

```
void BoyerMoore(string txt, string pattern){//12171820 이성인
    int txtLen = txt.length(); // 길이를 받습니다.
    int patternLen = pattern.length(); // 길이를 받습니다.
    int skip[256] = { -1, }; // 모든 배열을 -1로 초기화한다
    for(int i=0 ; i<patternLen ; i++)
    {
        skip[(int)pattern[i]] = i; // map과 유사하게 사용하는 경우입니다.
        // (int)pattern[i]에 해당하는 값을 받아서 그 문자를 찾으며 동시에 초기화 시킵니다.
        // 초기화 된 값을 skip을 통해서 이동합니다.
    }
```

Txt 와 pat를 받아서 길이를 저장합니다.

그후 skip될 배열을 모두 초기화 한 후, map과 유사하게 (int)pat[i]에 해당하는 int 값으로 바꾸어 skip 배열에 저장합니다. 이렇게 저장하는 경우 skip 배열을 돌면서 매번 식별할 문자를 찾지 않아도 됩니다. map 과 유사합니다.

```
int s = 0, j; // s는 text에 대한 patterntern의 이동값
//j는 현재 위치에서 이동하는 것을 의미
//12171820 이성인
while(txtLen-patternLen >s){
    j = patternLen-1;

    while(pattern[j] == txt[s+j] && j >= 0 )
    j--; // patterntern의 문자와 text가 매칭하면 j를 줄여 틀릴때까지 이동합니다.
        //  문자를 찾는다
    if(j < 0) // 문자열이 일치하였을 때
    {
        cout << "patterntern's index : " << s <<endl;
        s += (s>=txtLen-patternLen)? 1: patternLen-skip[(int)txt[s+patternLen]] ;
    }
    else{ // 문자열이 일치 하지 않았다면 이동합니다.
        s += max(1, j-skip[(int)txt[s+j]]); // 본문의 s 의 이동과 같습니다. if(M-j > k)를 비교하여
    }//12171820 이성인
}
}
```

While문은 해당 문자열의 끝에 도달할때까지 s를 눌려주면서 차후 종료합니다.

그후 while 문을 통해서 해당 j 위치에서 앞쪽 index 로 이동하면서 값이 달라 질때 까지 이동합니다. 차후 j가 -1이 되는 순간 문자열을 찾게 된것이며 해당 위치에서 j를 찾았다고 하며 차후 patern의 길이에서 skip만큼 점프해줍니다.

3. 실행 화면

다음과 같은 최적의 경로를 거쳐서 403이라는 결과가 나오게 되었습니다.

- Similar의 결과

```
=c++14 main.cpp -o main && "/Users/se
pattern's index : 1827
pattern's index : 14732
pattern's index : 16501
pattern's index : 45091
pattern's index : 45384
seong-in@iseong-in-ui-MacBookAir 중간
```

- UA -----v 검색 결과

```
=c++14 main.cpp -o main && "/Users/seong-
pattern's index : 11347
pattern's index : 12659
seong-in@iseong-in-ui-MacBookAir 중간과제
```

- HTTP/1.1, 검색결과

```
=c++14 main.cpp -o main && "/Users/seong-
pattern's index : 14392
```

- “letter "A".."Z" 검색결과

```
//12171820 이성인
    string pat = "letter \"A\"..\"Z\"";
    BoyerMoore(txt, pat);
```

```
=c++14 main.cpp -o main && "/Users/seong-in/De
pattern's index : 18357
seong-in@iseong-in-ui-MacBookAir 중간과제 %
```

- “[50]” 검색결과

```
=c++14 main.cpp -o main && "/Users/seong-in/De
seong-in@iseong-in-ui-MacBookAir 중간과제 %
```

4. 결론

대표적인 문자열 탐색 알고리즘인 보이어 무어 알고리즘을 이용하여 문자열 탐색을 진행했습니다. File 의 크기가 크다 보니 직선적 알고리즘을 사용시 큰 시간이 걸리게 됩니다. 보이어 무어 알고리즘을 이용하여 효과적으로 찾고자 하는 문자열을 찾을 수 있었습니다.