**Algorithm W7 Assignment**

**2015003654 Lee Hoyoung**

**Horspool Algorithm**

1.소스코드 설명

int table[500];

매칭이 안되었을 시 이동할 칸 수를 저장할 테이블

int horspool(int \*text, int \*pattern, int tLength, int pLength)

text: 패턴을 확인할 텍스트

pattern: 패턴값

tLength: text의 길이

pLength: pattern의 길이

함수 내에서 계산 시 sizeof(pattern)/sizeof(int) 값이 이상한 값이 나와서 아예 매개변수로 길이까지 받아버렸다.

테이블 값을 모두 pLength로 초기화함, pattern에 따라서 패턴 내 존재하는 알파벳 값들만 table 값을 바꿔주었다.

패턴과 텍스트의 단어가 매칭되었을 때는 그 단어가 시작하는 index를 반환한다.

텍스트에 패턴과 매칭되는 단어가 없다면 -1을 반환한다.

2.소스 결과

int text[] = {'T', 'R', 'U', 'T', 'H', 'L', 'E', 'A', 'D', 'S', 'Y', 'O', 'U', 'T', 'O', 'F', 'R', 'E', 'E', 'D', 'O', 'M'};

    int pattern1[] = {'T', 'R', 'U', 'T', 'H'};

    int pattern2[] = {'F', 'R', 'E', 'E', 'D', 'O', 'M'};

 result = horspool(text, pattern1, sizeof(text) / sizeof(int), sizeof(pattern1) / sizeof(int));

    printf("%d번째에서 pattern이 발견됨\n", result);

    result = horspool(text, pattern2, sizeof(text) / sizeof(int), sizeof(pattern2) / sizeof(int));

    printf("%d번째에서 pattern이 발견됨\n", result);

0번째에서 pattern이 발견됨

15번째에서 pattern이 발견됨

결과를 확인해보면 맞게 나온다.

3.참고자료 및 구현 환경

코드 구현엔 수업자료를 참고하라고 하셔서 수업자료의 pseudo code를 참고하였고 visual studio code 1.51.1 버전에서 컴파일하였습니다.

C 파일을 새로운 파일에서 컴파일하여도 동작합니다.