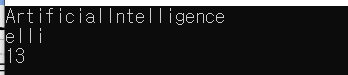
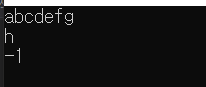
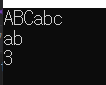
HW2 KMP 알고리즘

20161637 장호영



<코드 실행 결과>

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

char failure[1024];

void func\_fail(char\* pat) {

int n = strlen(pat);

int i = -1, j = 0;

failure[0] = -1;

while (j < n) { //패턴 길이동안 반복

if (i == -1)

failure[++j] = ++i; //첫 시작

else if (i >= 0 && pat[i] == pat[j])

failure[++j] = ++i; //패턴이 맞으면 갱신

else i = failure[i]; // 패턴 매칭 실패 시

}

}

int pmatch(char\* str, char\* pat) {

int i = 0, j = 0;

int lens = strlen(str);

int lenp = strlen(pat);

while (i < lens && j < lenp) { //비교할 문자열 있으면

if (str[i] == pat[j]) { //일치했을 경우

i++;

j++;

}

else if (j == 0) // 일치하지 않았지만 패턴 시작부분일 경우

i++;

else

j = failure[j - 1] + 1; //일치하지 않았을 경우

}

return ((j == lenp) ? (i - lenp) : -1);

}

int main() {

char str[1024];

char pat[1024];

scanf("%s", str);

scanf("%s", pat);

func\_fail(pat); //failure function 만들기

printf("%d\n", pmatch(str, pat)); //pmatch로 패턴 위치 파악

return 0;

}

1. 먼저 전역으로 failure function을 위해 char failure[1024] 선언
2. void func\_fail(char\* pat)은 입력받은 pattern을 매개변수로 하여 failure function을 만들어 나중에 최대한 뒷걸음질을 적게할 수 있도록 맞는 패턴의 시작 부분을 기억할 수 있도록 해주는 함수
3. int pmatch(char\* str, char\* pat)는 문자열과 패턴을 매개변수로 하고 전역 변수인 failure function을 이용, str과 pat이 모두 끝까지 확인할 것이 없을 때까지 돌면서 문자열과 패턴을 비교하여 패턴 발견시 시작 위치 반환하고 발견이 안되면 –1 반환
4. main에서는 위 함수 호출