자료구조\_과제3

String Matching

본 과제는 C++을 기반으로 작성하였습니다.



목 차

1. Naive Algorithm

2. KMP Algorithm

자료구조

김성열교수님

소프트웨어학과

201811182

박원준

2019.04.02

2. KMP Algorithm

#include <string>

#include <vector>

#include <iostream>

using namespace std;

string T, P; // T : text, P : pattern

vector<int> FailureFunction()

{

int pattern\_size = P.size();

vector<int> table(pattern\_size, 0);

int j = 0;

for (int i = 1; i < pattern\_size; i++)

{

while (j > 0 && P[i] != P[j]) // j가 0이면, 탐색할 필요가 없음.

j = table[j - 1];

if (P[i] == P[j])

table[i] = ++j;

}

return table;

}

int main(void)

{

cin >> T >> P;

vector<int> table = FailureFunction();

int j = 0;

for (int i = 0; i < T.size(); i++)

{

while (j > 0 && T[i] != P[j])

j = table[j - 1];

if (T[i] == P[j])

if (j == P.size() - 1)

{

cout << i - P.size() + 2 << "번째에서 찾았습니다." << endl;

j = table[j];

}

else

j++;

}

return 0;

}