

# COMP319 Algorithms 1, Fall 2020

## Homework Programming Assignment 5 (HW5), 10%

### 알고리즘1, 2020년 가을 학기, 프로그래밍 숙제 5, 10%

Instructor: Gil-Jin Jang Email: gjang@knu.ac.kr  
School of Electronics Engineering, Kyungpook National University  
장길진, 경북대학교 전자공학부

프로그래밍 숙제 5번의 목적은 다음과 같다.

1. LCS (longest common subsequence) 알고리즘을 직접 구현하여 보고 이해한다.
2. Circular shift를 허용한 LCS 의 변형을 구현한다.

공통 요구사항:

1. `int main(int argc, char *argv[]) { ... }` 로 `main` 함수가 작성되어야 하며, 컴파일 명령에 따라 `./hw5.exe`, 또는 `./hw5.out`로 실행된다.
  - Linux 에서 `gcc -Wall hw5.c -o hw5.exe` 또는 `gcc -Wall hw5.c -o hw5.out` 명령으로 컴파일 가능하다. 여기서 `-Wall` 은 모든 compile warning 을 표시하게 한다.
2. 처음의 주석문에는 본인의 정보가 기재되어야 함.(10%)
3. 다른 학생의 코드와 거의 동일하면 COPY 판정을 받을 수 있기 때문에, 본인이 공개된 코드를 참조했으면 웹주소, 책이름, 강의자료 차수와 쪽수 등을 주석문으로 넣는다. 만약 다른 학생의 코드와 유사한데 참조정보가 없으면 COPY 판정을 받게 된다.
  - 특히 LCS 코드는 아주 쉽게 구할 수 있다. 조금이라도 인터넷 검색에서 정보를 얻었으면 안전하게 참조정보를 넣는 것이 좋다. LCS는 직접 작성하더라도 추가점은 거의 주어지지 않음
4. 한글문자는 코드에 넣지 않는다.
  - 한글 사용 자체에 대한 감점은 없지만 인코딩 방식이 다르면 컴파일 오류를 발생시킬 수 있으며, 실행점수를 전혀 못 받을 수도 있음. 이름도 영어로 적는다.
  - 한글 문자를 주석 등에 사용하고자 하는 경우 linux machine 에서 컴파일해 보고 확인하거나, UTF8 로 인코딩을 바꾸어 제출한다(안전하게는 안 쓰는 편이 좋다).
5. 정확한 메모리 사용량을 측정하기 위해서는 배열을 사용할 경우에는 `malloc.c()` 함수를 사용하고, 문자열을 복사하여 사용할 경우 `strdup` 대신 `strdup.c` 함수를 사용한다.

`malloc.c(size_t size)` 전역 변수 `used_memory` 에 size 를 누적하고 `malloc`을 호출

`strdup.c(const char *s)` 전역 변수 `used_memory` 에 s의 길이(`strlen(s)+1`)를 누적하고 `strdup` 호출

**NOT ALLOWED 1** 이 두 함수만을 이용하여 배열 사용이 허용됨. 배열이 아닌 일반 변수는 사용량에서 무시한다.

**NOT ALLOWED 2** `malloc`, `strdup` 직접 사용하거나 `calloc`, `realloc` 등 다른 메모리 할당 함수는 사용하지 않는다(감점요인).

**NOT ALLOWED 3 memcpy, memccpy, memmove, wmemmove** 등의 함수는 사용하지 않는다. (cheating으로 간주되어 시간 점수 0점이 부여될 수 있음) 이 함수들은 시스템 HW call 등을 이용하여 매우 효율적으로 구현되어 있으며 UNIT OPERATION TIME 가정에 위배되기 때문에 제대로 평가를 할 수 없다.

출력된 메모리 사용량은 TA가 코드를 보고 누락된 것이 없는지 검증한다. 필요시 코드를 수정하거나, 수작업으로 값을 업데이트할 수 있으며 작성한 코드가 너무 복잡하여 메모리 사용량을 알 수 없을 경우 '판독불가'로 평가하고 점수를 부여하지 않는다.

6. 참고: 두 파일의 차이를 비교할 때에는 windows 에서는 winmerge (<http://winmerge.org>), linux 에서는 vimdiff 사용 가능.

- vi/vim 에 익숙한 사용자는 쉽게 사용법을 알 수 있다.
- 용법: vimdiff a.c b.c

## 1 구현해야 할 LCS 방법들

### 1.1 LCS (longest common subsequence) 의 정의

문자열(string/character sequence)에 subsequence 란 일부의 character 를 원래의 상대적인 순서를 유지하여 선택한 문자열을 의미한다. 즉,

$\text{subsequence}(\text{"CCGTGTAT"}) = \{\text{"C"}, \text{"CCG"}, \text{"CGTA"}, \text{"CAT"}, \dots\}$

두 개의 문자열이 주어졌을 때, 공통되는 subsequence 중에서 가장 긴 것을 LCS (longest common subsequence) 로 정의할 수 있다.

$\text{LCS}(\text{"TGC**GA**TGA"}, \text{"GT**CC**AT**GT**"} ) = \text{"**TCATG**"}$

### 1.2 LCS with circular shift 의 정의

Circular shift 란 sequence 의 각각의 원소를 주어진 shift 수만큼 뒤쪽으로 이동시키고, sequence 의 바깥으로 나오는 원소는 다시 앞으로 붙이는 shift 방법을 의미한다. 즉,

$\text{circularshift}(\text{"CCGTTTCCA"}, 1) \rightarrow \text{[ ] CCGTTTCC A} \rightarrow \text{ACCGTTTCC}$

$\text{circularshift}(\text{"CCGTTTCCA"}, 4) \rightarrow \text{[ - - - ] CCGTT TCCA} \rightarrow \text{TCCACCGTT}$

두 개의 문자열이 주어졌을 때, 공통되는 circular shift를 허용하는 subsequence 중에서 가장 긴 것을 LCS-CS 로 정의하고, LCS 결과와 비교하면 다음과 같다.

$\text{LCS}(\text{"**CCGTTTCCA**"}, \text{"TG**CC**AT**CT**"} ) = \text{"**CCTT**"}$

$\begin{aligned} &\text{LCS-CS}(\text{"CCGTTTCCA"}, \text{"TGCCCATCT"}) \\ &= \text{LCS}(\text{circularshift}(\text{"**CCGTTTCCA**"}, 4), \text{"TGCCCATCT"}) \\ &= \text{LCS}(\text{"**TCCA**CCGTT"}, \text{"TGCCCATCT"}) = \text{"**TCCACT**"} \end{aligned}$

(문자열은 하나만 shift하면 된다.)

이때, 만약 여러 개의 shift가 같은 길이의 LCS를 가진다면, 가장 작은 shift 값을 선택한다.

## 2 Homework 5

**할것1:** command-line arguments 로 주어진 두 개의 문자열의 LCS를 찾고 화면에 출력한다.

**할것2:** 같은 문자열에 대하여 LCS-CS를 찾고 화면에 출력한다.

**실행에 및 자세한 설명:** 채점에 사용되는 command line arguments 는 다양한 길이와 종류의 문자열이며, 길이의 제한이 없다. 파일입출력은 사용하지 않는다.

- (a) line 1, (argv[1]): 입력 문자열 1
- (b) line 1, (argv[2]): 입력 문자열 2
- (c) line 2, 화면출력줄1: 두 문자열의 LCS = GCA
- (d) line 3, 화면출력줄2: 두 문자열의 LCS-CS = GCA. 첫번째 예제는 LCS와 LCS-CS가 같음. 따라서 shift=0
- (e) line 4, 화면출력줄3: **0.00001 seconds** LCS, LCS-CS의 측정된 연산 시간
- (f) line 5, 화면출력줄4: **408 bytes ( 16.320 x 25 )** 측정된 사용 메모리 동적 메모리로 사용된 메모리의 byte 수와 입력크기(두 개의 문자열의 길이의 합)의 배수로 표현됨
- (g) line 13, 입력 문자열 1 = CCGTTTCCA, 입력 문자열 2 = TGCCCATCT
- (h) line 14: 두번째 예제의 LCS — CCTT. 길이는 3
- (i) line 15: 두번째 예제의 LCS-CS — TCCACT. 길이는 6, shift=4

```
1      $ ./hw5.exe GCAAA GGGCA
2      1) S1 S2 LCS = GCAAA GGGCA GCA
3      2) S1(shift=0) S2 LCS = GCAAA GGGCA GCA
4      0.00001 seconds
5      408 bytes ( 16.320 x 25 )
6
7      $ ./hw5.exe CCAAC CCGTGTAT
8      1) S1 S2 LCS = CCAAC CCGTGTAT CCA
9      2) S1(shift=0) S2 LCS = CCAAC CCGTGTAT CCA
10     0.00000 seconds
11     555 bytes ( 13.875 x 40 )
12
13     $ ./hw5.exe CCGTTTCCA TGCCCATCT
14     1) S1 S2 LCS = CCGTTTCCA TGCCCATCT CCTT
15     2) S1(shift=4) S2 LCS = TCCACCGTT TGCCCATCT TCCACT
16     0.00001 seconds
17     1000 bytes ( 12.346 x 81 )
18
19     $ ./hw5.exe GCTTC CGCTCTA
20     1) S1 S2 LCS = GCTTC CGCTCTA GCTT
21     2) S1(shift=1) S2 LCS = CGCTT CGCTCTA CGCTT
22     0.00001 seconds
23     506 bytes ( 14.457 x 35 )
24
25     $ ./hw5.exe AACTT CTGAGTA
26     1) S1 S2 LCS = AACTT CTGAGTA CTT
27     2) S1(shift=3) S2 LCS = CTAA CTGAGTA CTTA
28     0.00001 seconds
29     506 bytes ( 14.457 x 35 )
```

**template 코드 및 요구조건:** template\_hw5.c, examples\_hw5.log

- 1. "FILL" 로 표시가 되어 있는 영역에 코드 구현, 추가 함수 작성은 허용됨
- 2. 출력결과만 동일하면 약간 수정하여도 상관없다

**제출물:** hw5.c 오류 없이 컴파일 가능, 위의 실행예에 따라 실행가능해야 하며, 위의 예제의 형식에 맞는 출력을 화면에 보여야 한다.

**채점기준 및 평가방법:** Homework 5의 평가총점은 100점이며, 전체 성적에 10% 반영됨

**10점** 제출 기본점수

**10점** 학번/이름 주석문에 정확히 입력

**40점** 컴파일 오류 없고 실행이 정확한지 평가

- 예제로 주어진 입력과 다른 입력들에 대하여 정확한지 평가. 길이에는 제한이 없음. 각 예제별로 정답과 완전히 동일해야 한다 (한 글자도 틀리면 그 입력에 대하여 0점. line break 차이는 허용함. 출력들은 모두 공란 하나 ' '로 구분됨 )
- 컴파일 오류시 실행불가로 0점이 주어짐
- 답이 맞더라도 알고리즘을 제대로 구현하지 않으면 해당과제를 하지 않은 것으로 판단하여 실행점수 0점

**10점** 실행시간 점수(차이가 크지 않을 것으로 예상됩니다)

- 10: 다른 학생들과 거의 비슷하거나 심하게 느리지 않음
- 0: 너무 오래 걸림

주관적이지만 알고리즘을 제대로 구현하였을 경우, 즉 대부분의 학생들이 만점을 받도록 분포를 보고 결정하겠습니다.

**10점** 메모리 사용량 점수(차이가 크지 않을 것으로 예상됨)

- 10: 다른 학생들에 비해 월등하게 사용량이 적음
- 5: 다른 학생들과 거의 비슷함
- 0: 메모리를 너무 많이 사용함

출력된 메모리 사용량은 TA가 코드를 보고 누락된 것이 없는지 검증한다. 필요시 코드를 수정하거나, 수작업으로 값을 업데이트할 수 있으며 작성한 코드가 너무 복잡하여 메모리 사용량을 알 수 없을 경우 '판독불가'로 평가하고 점수를 부여하지 않는다.

**20점** 코드에 대한 평가(조건들 만족)

**COPY** 적발시 전체 0점. COPIED/BEING COPIED 모두 해당됨.

### 3 제출형식 및 방법

**제출할 코드** hw5.c 만을 제출해야 함. 필요없는 파일 제출시 10점까지 감점 있음

**제출방법** 위의 파일을 hw5.zip 을 만들고, lms.knu.ac.kr 에 업로드한다. LMS는 제출된 파일의 이름을 복잡하게 바꾸기 때문에 개별파일을 제출하면 채점이 매우 어렵다.

**Due** 11/27 금요일 11:59 LMS time

**Late submission** 11/28 토요일 09:59 LMS time, 시간당 10점 감점