

COMP319 Algorithm 1, Spring 2020

Programming assignment 5 (총 3 문제)

Instructor: 장길진 (gjang@knu.ac.kr, Tel 053-950-5517)

TA: 박진희 (pjhdrrm@gmail.com, Tel 053-940-8617)

General information

1. 채점기준:

- a. **첫출의 주석으로 학번과 이름 제대로 명시: 20%**
 - i. 제대로 제출한 학생에게 기본점수를 주기 위함입니다. 꼭 쓰도록 하세요.
 - ii. 개발자를 명시하는 것은 지식재산권 보호를 위해 꼭 필요합니다. 습관처럼 적어야 합니다.
- b. 정확도: 30% - 주어지지 않은 다른 예제를 돌려보고 출력파일들의 정확도 평가
- c. 실행시간: 20% - 전체 실행시간을 측정하여 상대 평가.
 - i. 제출한 코드가 compile 이 안 될 경우 실행불가 → 정확도와 실행시간 점수(총 50%) 0 점
 - ii. 5-1 LCS 는 비정상적으로 시간이 오래 걸리지만 않으면 만점
 - iii. 답이 틀리면 실행시간 평가가 무의미하기 때문에 0 점. 따라서 정확도 우선으로 작성함.
- d. 코드 reading 점수 30% - 문제에서 요구하는 조건들을 만족시켰는가?
 - i. 문제에서 구현하라는 것을 무시하고 단순히 출력만 같을 경우 정확도 점수에서 추가감점

2. 감점기준

- a. 지각제출은 1 시간마다 10% 감점이 있습니다.
 - i. LMS 기준 00:00 ~ 00:59 -10%, 01:00 ~ 01:59 -20%, ..., 09:00 ~ 09:59 -100%
- b. Cheating (카피):
 - i. 교수/조교/튜터가 같이 보고 평가하여 다수결로 결정
 - ii. 동일한 코드가 나오면 "copied"/"being copied" 여부에 관계없이 해당 점수 0 점
 - iii. Internet 에서 찾거나 책에 있는 코드를 사용하였을 경우 제출한 코드 안에 주소나 책의 쪽수 등을 명시할 것. 자신이 참조한 코드의 정보를 명시하면 완전히 동일하더라도 cheating 으로 판단하지 않습니다. 즉, 많이 알려진 코드라면 출처를 명시하면 다른 학생들과 같아도 상관없습니다.
 - iv. 출처가 '친구', '선배' 등은 허용하지 않습니다.

3. 제출물

- a. C 소스코드(.c files)와 리포트(.pdf)만을 제출합니다.
- b. 필요없는 다른 파일을 제출하면 10%까지 감점이 있습니다.
- c. 소스 코드들과 리포트들을 하나로 묶어서 hw5.zip 파일로 제출함. 파일들을 따로 제출하면 채점이 매우 어렵습니다.
- d. Template 코드는 없습니다. 입출력만 적합하게 구현하면 됩니다.

4. Compilation:

- a. 실행기준: linux gcc 7.5.0 또는 code::blocks with MinGW gcc 5.4.0 under Windows
- b. Linux gcc 7.5.0 으로 먼저 평가하고, 오류가 있으면 code::blocks + gcc 5.4.0 로 재평가합니다. 주석으로 사용한 gcc 버전을 명시해 주면 채점에 도움이 됩니다.
- c. MniGW gcc 5.4.0 은 <https://files.1f0.de/mingw/> 에서 받을 수 있습니다.

문제 설명

1. (20%) Longest common subsequence (LCS)

text 입력에 대하여 LCS 를 구하는 C 프로그램을 작성한다. 찾은 LCS 와 dynamic programming table 이 채워지는 과정을 보인다.

입력예: ("fscanf(stdin, "%s"...)"로 받으면 됨)

```
ABCB BDCAB
```

출력예: (alignment 는 안 맞아도 크게 상관없고 개수만 맞으면 됨. 단, 숫자/문자 간 한 칸 이상 공백. 출력값은 강의자료 참고)

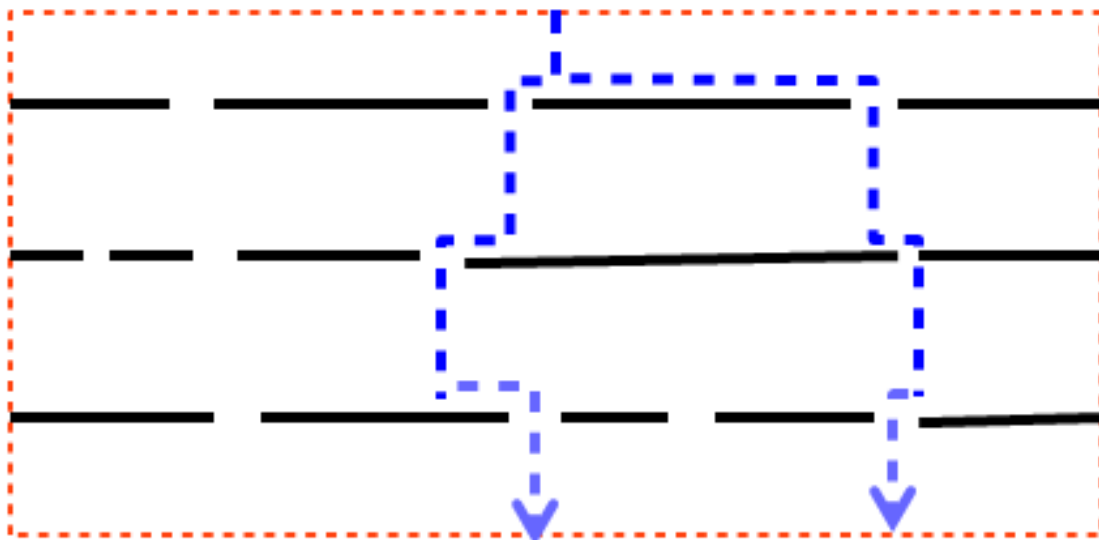
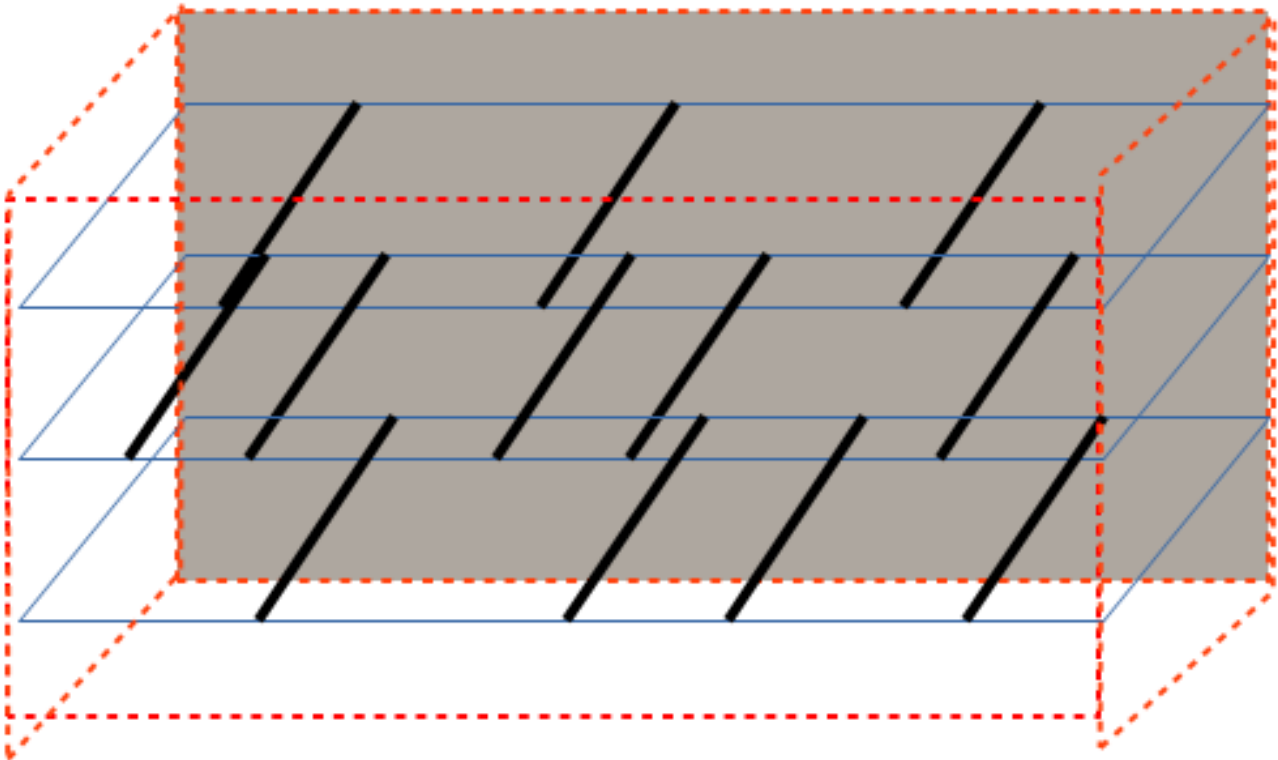
```
LCS BCB
      B D C A B
      0 0 0 0 0 0
A    0 0 0 0 1 1
B    0 1 1 1 1 2
C    0 1 1 2 2 2
B    0 1 1 2 2 3
```

요구조건(requirements)

- ① **Submission file (제출할 파일): lcs.c**
- ② 문제를 간단하게 하기 위하여 입력파일에는 숫자, 영어 대/소문자(case-sensitive)만 포함된다.
- ③ 최종출력은 화면에(sprintf(), fprintf(stdout, ...) 사용)
- ④ 주어지지 않은 예제로 평가함. 따라서 위의 예제 이외의 다른 예제도 돌려보아야 함

2. (40%) Shortest water slit path

A box is partitioned by plates with slits (very thin gaps), and they are equally distant from each other vertically. Suppose water is poured from the center of the top, and the water will flow down through the slits, to the base. Design an algorithm that finds the minimum distance that water travels, and the slit numbers on the minimum path. See the following figure for details.



To do: Write a C code that finds

- minimum distance from the top-center to at any point of the bottom. Note that all the vertical distances are the same.
- slit locations at each plate that the minimum path goes through.

입력예: ("fscanf(stdin, "%s"...)"로 받으면 됨)

문제를 간단하게 하기 위하여 모든 수치는 양의 정수로 주어지며, 음수값은 입력을 구분하는데 사용된다. 이를 이용하여 입력에서 줄바꿈을 인식할 필요가 없다. (줄바꿈은 편의상으로만 주어짐)

Plate 의 개수 plate 의 폭 **-1**

첫번째 plate 의 slit 의 개수와 위치들 **-2** (첫번째 plate 가 끝났음을 알림)

두번째 plate 의 slit 의 개수와 위치들 **-2** (두번째 plate 가 끝났음을 알림)

...

-3 (전체 입력이 끝났음을 알림)

실제 입력은 다음과 같다.

Plate 의 수는 3 개, 폭은 10,

첫번째 plate 는 3 개의 slit 이 있고 그 위치는 2, 6, 9,

두번째 plate 는 4 개의 slit 이 있고 그 위치는 1, 3, 5, 8

```
3 10 -1
3 2 6 9 -2
4 1 3 5 8 -2
2 7 9 -2
-3
```

또는 아래와 같이 줄바꿈을 없애도 동일하다.

```
3 10 -1 3 2 6 9 -2 4 1 3 5 8 -2 2 7 9 -2 -3
```

출력예: 최소거리는 4, 각각의 slit 위치는 6, 5, 7 이 선택됨을 의미함

```
4 6 5 7
```

요구조건(requirements)

- ① **Submission file (제출할 파일. 소스코드와 리포트): sws.c, sws.pdf**
- ② **sws.pdf:** 본인이 사용한 알고리즘을 A4 1 쪽 이내로 간단하게 설명. Code reading 점수에 포함
- ③ 최종출력은 화면에(sprintf(), fprintf(stdout, ...) 사용)
- ④ 주어지지 않은 예제로 평가함. 따라서 위의 예제 이외의 다른 예제도 돌려봐야 함

3. (40%) 제한된 경로의 길이로 최대한 많은 점 방문하기(forward travelling salesperson)

비행기를 타고 도시를 방문하는 판매원이 있다. 연료의 양에 제한이 있고 시작점과 끝점이 주어졌을 때, 가장 많은 수의 도시를 방문하는 경로는? 단, 2 차원 평면에서 x 축으로는 전진만이 가능하다

입력:

- (고정값) 100 x 100 지도
- (고정값) 연료의 양은 4 가지 300, 700, 1100, 1500
- (고정값) 시작 도시 좌표 (0, 0), 마지막 도시 좌표 (100, 100). 정렬되어 있지 않음
- (변수) 98 개의 도시의 (x, y) 좌표, $1 \leq x \leq 99$, $1 \leq y \leq 99$

```
100
0 0   (첫번째 도시 x y 좌표)
36 37 6 21 52 9 34 77 ... (2~99 번째 도시 x y 좌표들. 생략됨. 첨부된 입력파일 참조)
100 100 (마지막 도시 x y 좌표)
```

출력:

- 주어진 거리, 실제 거리, 방문한 도시들의 좌표
- 아래 값들은 예시이고 실제 답은 아니다. [...] 은 생략
- 300, 700, 1100, 1500, 4 가지 연료에 대해 모두 출력

```
300 279.6
0 0 ... 100 100 (중간도시는 생략됨)

700 691.1
0 0 2 43 ... 100 100 (중간도시는 생략됨)

1100 1099.9
0 0 2 43 6 21 ... 97 76 100 100 (중간도시는 생략됨)

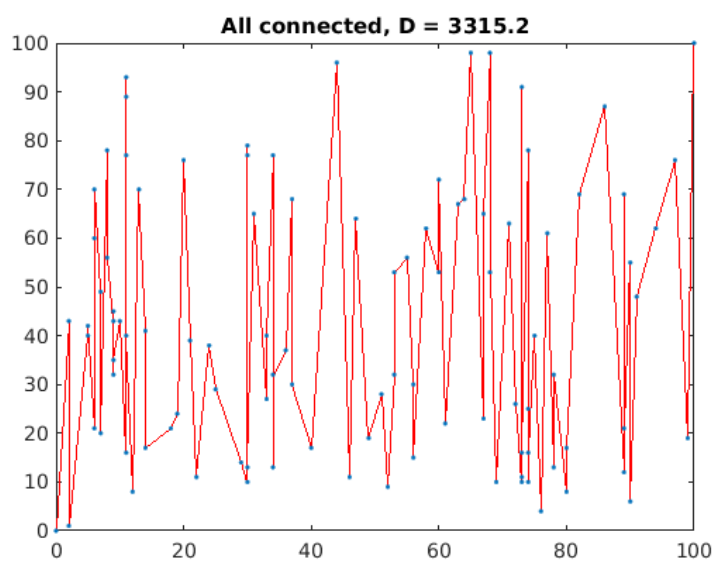
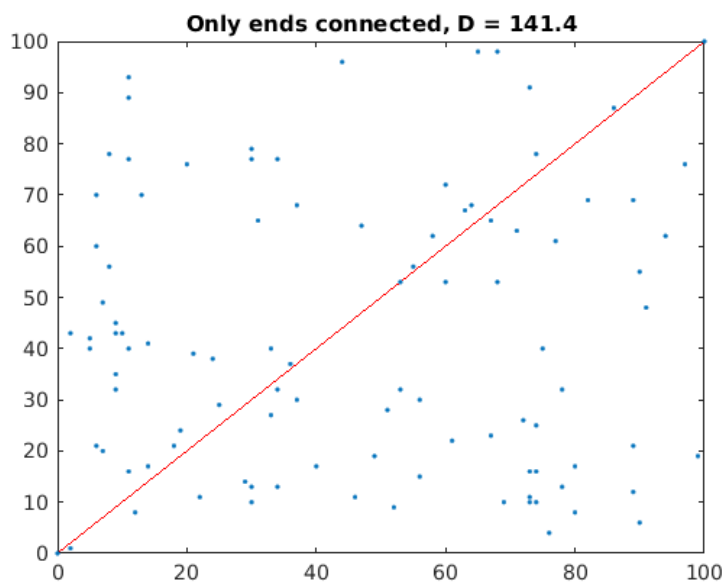
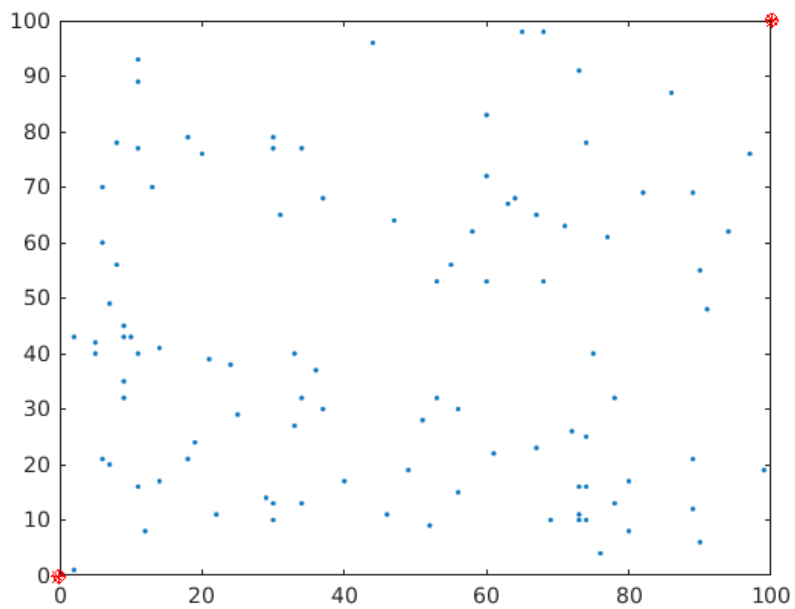
1500 1497.2
0 0 2 43 5 40 6 21 ... 97 76 100 100 (중간도시는 생략됨)
```

NOTES:

- x 축으로는 전진만이 가능
- x 축 좌표가 같은 도시가 있을 수도 있다(방문 순서 바뀔 수 있다)
- x 축과 y 축 좌표가 둘다 같은 도시가 있을 수도 있다(복수로 센다)

요구조건(requirements)

- ① **Submission file (제출할 파일. 소스코드와 리포트):** [fts.c](#), [fts.pdf](#)
- ② **fts.pdf:** 본인이 사용한 알고리즘을 A4 1 쪽 이내로 간단하게 설명. Code reading 점수에 포함
- ③ 최종출력은 화면에(`printf()`, `fprintf(stdout, ...)` 사용)
- ④ 주어지지 않은 예제로 평가함. 따라서 위의 예제 이외의 다른 예제도 돌려봐야 함



(끝)