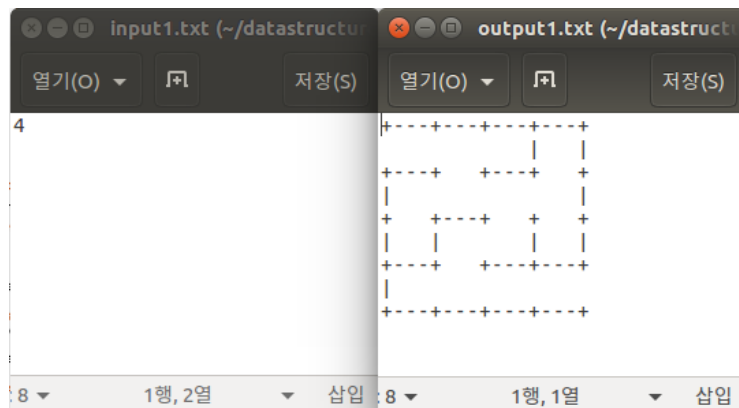


## Lab 07 (4/20) Maze

- arranged by TA Sehyun Cha

**Due : ~2023.04.26.(Wed) 23:59, Late Submission : ~2023.04.27.(Tue) 23:59**

- **Random Maze Implementation using Disjoint Set.**
- 아래의 사진과 같이 [input].txt 파일에서 생성할 **Maze**의 크기를 입력받아 **Maze**를 생성하고, 생성된 **Maze**의 모양을 [output].txt 파일에 출력하여 저장한다.
- **Implement the function that creates Random Maze by receiving the [input].txt file as input as below, and the output results of generated Maze are stored in the [output].txt file.**



- **<input> :**  
Maze Matrix의 width, height size. (maze는 정사각 행렬, 크기는 100 이하의 자연수)  
**Width, Height size of the Maze(Matrix).** (maze is a square matrix, natural number less or equal to 100)
- **<output> :**  
**생성된 Maze 모양 출력.**  
**Print out the shape of the generated Maze.**

### <Structure & Function Format>

```
typedef struct _DisjointSet {  
    int size_maze;  
    int *ptr_arr;  
}DisjointSets;
```

```
void init(DisjointSets *sets, DisjointSets *maze_print, int num);  
void Union(DisjointSets *sets, int i, int j);  
int find(DisjointSets *sets, int i);  
void createMaze(DisjointSets *sets, DisjointSets *maze_print, int num);  
void printMaze(DisjointSets *sets, int num);  
void freeMaze(DisjointSets *sets, DisjointSets *maze_print);
```

- 위 사진과 같은 Struct 구조체를 사용하셔야 합니다.
- 위 사진과 같은 함수들을 형식에 맞게 구현해주시면 됩니다.
- **Struct format above should be used for implementation**
- **Functions should be implemented in appropriate format as above.**

### <init>

- set의 메모리를 할당하고 값들을 초기화 합니다.
- sets은 num\*num 개의 disjoint sets을 표현합니다.
- maze\_print는 num\*num\*2 개의 값을 가지고 있습니다.
- maze\_print의 각 원소는 해당 지점의 오른쪽, 아래 벽의 존재를 나타냅니다.
- Allocate memory and initialize sets.
- sets represent num\*num of disjoint sets.
- maze\_print have num\*num\*2 values.
- Each element of maze\_print represents the existence of the right and bottom walls at the corresponding point.

### <Union>

- 입력으로 받은 두 원소가 속한 Disjoint set을 합칩니다.
- Merge the Disjoint sets to which the two input elements belong.

### <Find>

- 입력으로 받은 원소가 속한 집합의 대표 원소를 반환합니다.
- Return the representative element of the set to which the input element belongs.

### <CreateMaze>

- 랜덤하게 엣지를 지워 cycle이 없는 미로를 만듭니다.
- 벽의 존재 유무는 maze\_print에 저장합니다.
- disjoint sets이 하나의 set이 될 때 까지 엣지를 지우면 됩니다.
- Build a maze without any cycle by choosing edges randomly.
- The existence of the wall is stored in maze\_print.
- Remove edges until the disjoint set becomes one set.

### <printMaze>

- **Maze** 출력시 :
  - Cell의 위치는 "공백3칸",
  - Horizontal Edge가 존재할때는 "---"(언더바가 아닌 마이너스 연산자 기호 3개),
  - Horizontal Edge가 존재하지 않을때는 "공백3칸",
  - Vertical Edge는 "|" (bitwise "or" 연산자 기호),
  - Vertical Edge가 존재하지 않을경우 "공백1칸",
  - 이외의 남은 공간은 "+" 로 표시하시면 됩니다.
- 단순히 **fprintf()** 함수만을 이용해서 구현한경우, **Lab07 0점처리 및 전체 주차별 과제점수에서 일정비율 감점이 이뤄질 예정입니다.**
- **Printing Maze:**
  - space for the Cell is "three space",
  - "---"(three minus operator (not the underbar "\_")) for Horizontal Edge,
  - "three space" when Horizontal edge does not exist,
  - "|" (bitwise "or" operator) for Vertical Edge,
  - "one space" when Vertical edge does not exist,
  - "+" for the rest.

- If it is implemented by just using the fprintf() function, Lab07 will be scored 0 and there will be disadvantages for the scores of the other lab assignments.

## <freeMaze>

- sets에 할당된 메모리를 해제합니다.
- Free allocated memory of sets

### <File Name Format>

- [StudentID].c      ex) 20XXXXXXXXX.c

## <Execution>

- gcc 20XXXXXXXX.c -o 20XXXXXXXX
- ./20XXXXXXXX [input\_file\_name] [output\_file\_name]

**<Issue>**

- 코드 작성시 주석을 적어주시기 바랍니다. 주석이 없는경우 Cheating으로 간주될 수 있습니다.
  - 제공된 testcase는 채점 case에 포함됩니다. 모두 알맞게 나오는지 확인해보시기 바랍니다.
  - 파일 입출력은 `argv[]` 를 사용하여 구현해주시기 바랍니다.
  - 제출 마감 시간 이전의 가장 최신 버전의 commit을 기준으로 채점할 예정입니다.
  - 제출 파일과, 폴더 naming 은 꼭 지정된 형식으로 해주셔야 합니다.
  - Please write down the detailed comments when writing the code. If there is no comment, it might be considered cheating.
  - Provided test case is included in the test case for grading. Please check to see if it makes a proper result.
  - Do not use a fixed file name when inputting and outputting files, but implement it using `argv[]` as in skeleton code.
  - Scoring will be based on the latest version of commit before the deadline.
  - The names of the .c file and directory should be named in proper format.
- 
- Random하게 edge를 지우기 때문에 example output과 결과가 매번 달라야 합니다.
  - Cycle 없이 단 하나의 start -> end path가 존재해야 합니다.
  - Since the edges should be eliminated randomly, output results after different execution with the same input value should be different every time.
  - Only one path from the start to end should exist without any cycle.

## <Directory Format>

- 아래와 같이 git 프로젝트 폴더에 "lab07" 폴더 생성후, "lab07" 폴더 안에 "20XXXXXXXXX.c" 파일을 위치시키시면 됩니다.
- After creating the "lab07" directory in the git-project-directory as below, place the "20XXXXXXXXXX.c" file in the "lab07" directory.

[illegible]

2023 CSE2010 20XXXXXXXXX/ (GitLab project directory)

\_\_\_\_\_

[illegible]