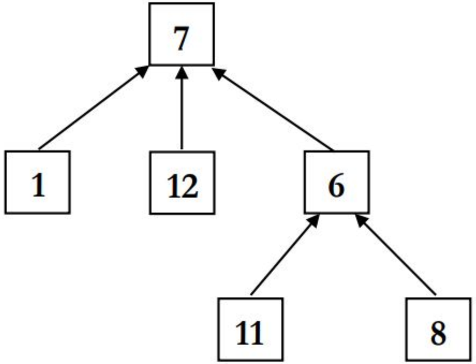
Lab 07: Maze(Disjoint Set)

Data Structure 2023

Disjoint Set

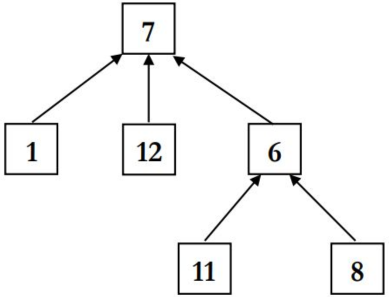
- If S_i and S_j are two sets and i != j, then there is no element that is in both S_i and S_j
- Maintain elements of S in a forest of inverted trees
  - pointers in the tree are directed towards the root.
  - the root of a tree has a NULL parent pointer
  - two elements are in the same set if they are in the same tree.

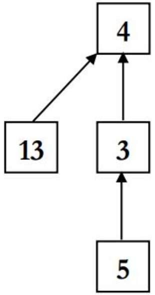


S_1 = {1, 6, 7, 8, 11, 12}

Disjoint Set

- Find(S, i)
  - find the node containing i
  - follow the parent links up to the root.
  - **return the root node** as the “name” of the set.

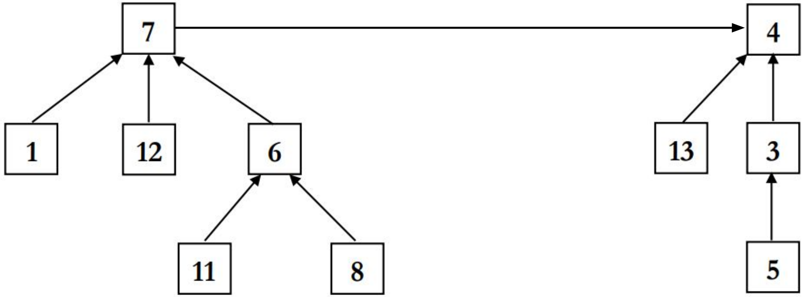




S_1 = {1, 6, 7, 8, 11, 12} **S_2 = {3, 4, 5, 13}**

Disjoint Set

- Union(i, j)
  - **link the root** of one tree into the root of the other tree

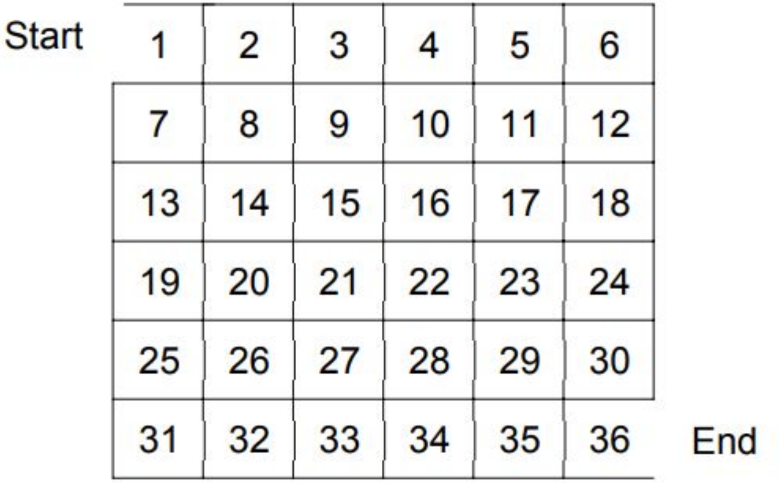


$$S_1 = \{1, 6, 7, 8, 11, 12\} \qquad S_2 = \{3, 4, 5, 13\}$$

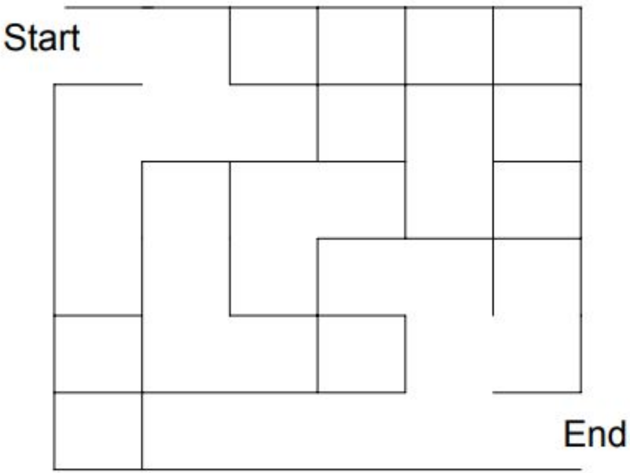
$$S_1 \cup S_2 = \{1, 6, 7, 8, 11, 12, 3, 4, 5, 13\}$$

Disjoint Set ADT

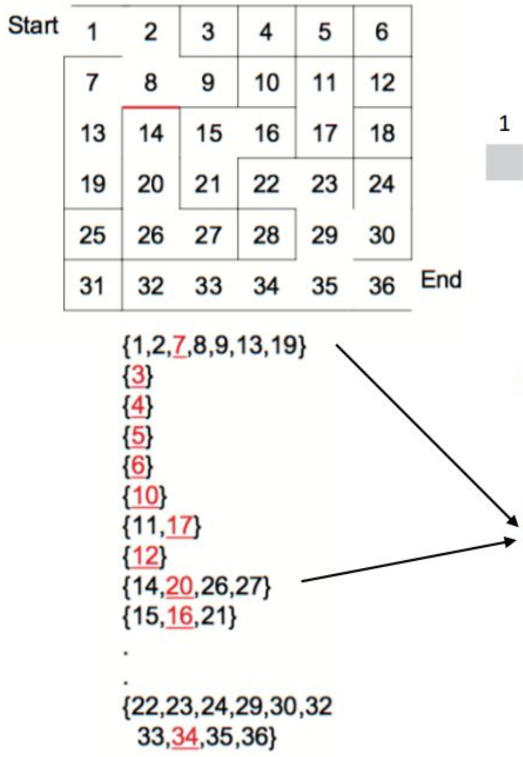
- Build a random maze by erasing edges.



1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36



Disjoint Set ADT



Start						
Start	1	2	3	4	5	6
	7	8	9	10	11	12
	13	14	15	16	17	18
	19	20	21	22	23	24
	25	26	27	28	29	30
	31	32	33	34	35	36

End

{1,2,7,8,9,13,19}
{3}
{4}
{5}
{6}
{10}
{11,17}
{12}
{14,20,26,27}
{15,16,21}
.
.
{22,23,24,29,30,32
33,34,35,36}

1	2	3	4	5	6	7	8	...	14	15	16	17	18	19	20	21	...
						-5	7		20						-3		

Union(8, 14)

Find(8) = 7, Find(14) = 20

{1,2,7,8,9,13,19, 14, 20, 26, 27}

Disjoint Set ADT

- **void init(DisjointSets *sets, DisjointSets* maze_print, int num)**
  - Initialize all sets to sets and maze_print
- **void union(DisjointSets *sets, int i, int j)**
  - Union two sets
- **int find(DisjointSets *sets, int i)**
  - Find the set including the number and return the representative member of the set
- **void createMaze(DisjointSets *sets, DisjointSets *maze_print, int num)**
  - Generate a maze that includes a path from Start position to End position **WITHOUT any cycles**
  - You can generate such a maze by randomly choosing a cell and direction. Use Union-Find ADT
  - For random number generation, use the library functions (**rand() in <stdlib.h>**)
- **void printMaze(DisjointSets *sets, int num)**
  - Print the resulting maze
- **void freeMaze(DisjointSets *sets, DisjoiuntSets *maze_print)**
  - Free memory of the maze

Disjoint Set ADT

Structure

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

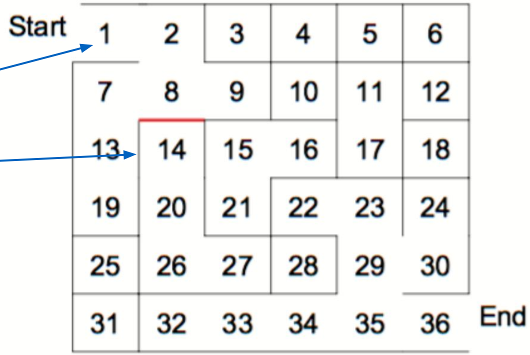
Function

```
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);
```

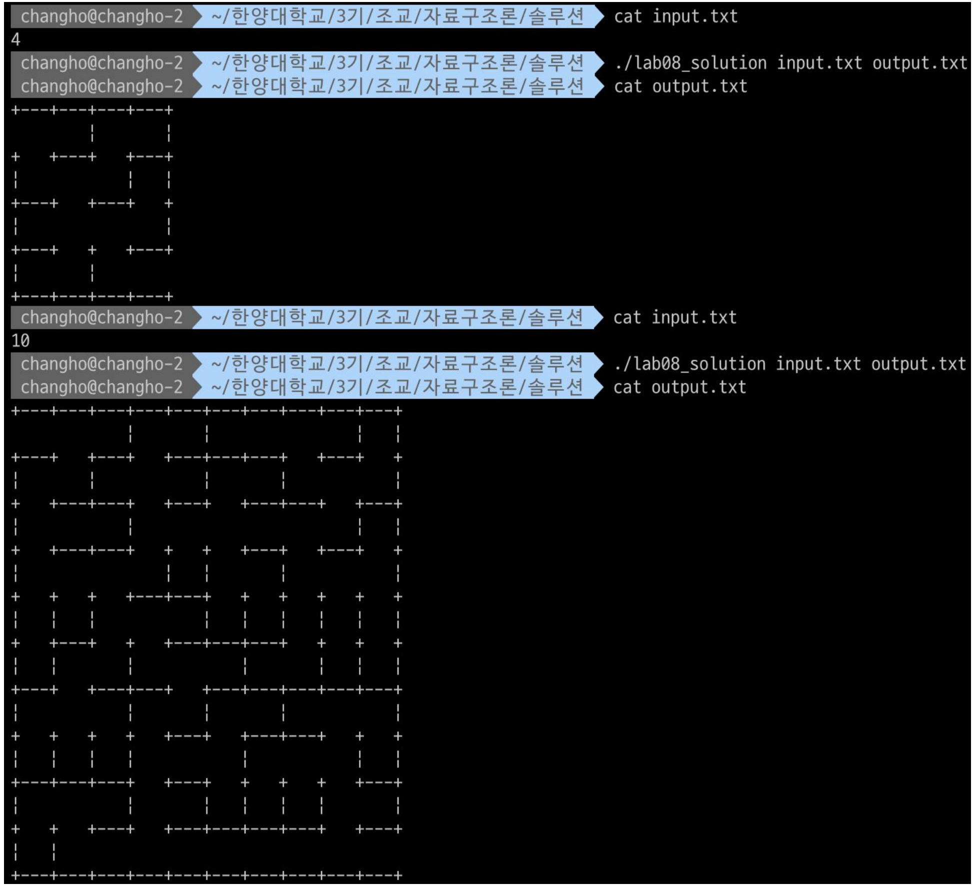
Variable

sets : means the number between the walls

maze_print : means the wall
1:yes, 0:no

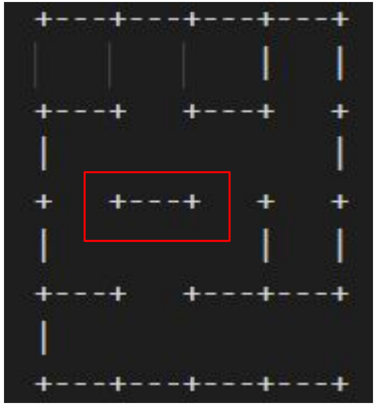


Input & Output Examples

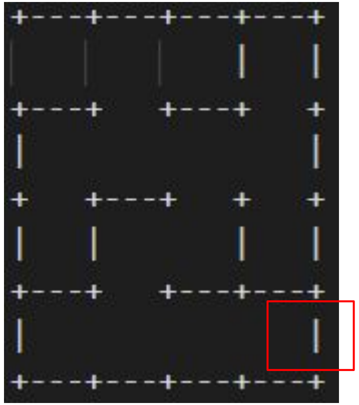


```
changho@changho-2  ~/한양대학교/3기/조교/자료구조론/솔루션  cat input.txt
4
changho@changho-2  ~/한양대학교/3기/조교/자료구조론/솔루션  ./lab08_solution input.txt output.txt
changho@changho-2  ~/한양대학교/3기/조교/자료구조론/솔루션  cat output.txt
+---+---+---+---+
        |       |
+   +---+   +---+
|           |   |
+---+   +---+   +
|               |
+---+   +   +---+
|       |
+---+---+---+---+
changho@changho-2  ~/한양대학교/3기/조교/자료구조론/솔루션  cat input.txt
10
changho@changho-2  ~/한양대학교/3기/조교/자료구조론/솔루션  ./lab08_solution input.txt output.txt
changho@changho-2  ~/한양대학교/3기/조교/자료구조론/솔루션  cat output.txt
+---+---+---+---+---+---+---+---+---+---+
            |       |               |   |
+---+   +---+   +---+---+---+   +---+   +
|       |           |       |           |
+   +---+---+   +---+   +---+---+   +---+
|           |                       |   |
+   +---+---+   +   +   +---+   +---+   +
|               |   |       |           |
+   +   +   +---+---+   +   +   +   +   +
|   |   |           |   |   |   |   |   |
+   +---+   +   +---+---+---+   +   +   +
|   |       |           |       |   |   |
+---+   +---+---+   +---+---+---+---+---+
|           |       |       |           |
+   +   +   +   +---+   +---+---+   +   +
|   |   |   |           |           |   |
+---+---+---+   +---+   +   +   +   +---+
|           |       |   |   |   |       |
+   +   +---+   +---+---+---+---+   +---+
|   |
+---+---+---+---+---+---+---+---+---+---+
```

Error Examples



```
+---+---+---+---+
|   |   |   |   |
+---+   +---+   +
|               |
+   +---+   +   +
|           |   |
+---+   +---+---+
|
+---+---+---+---+
```



```
+---+---+---+---+
|   |   |   |   |
+---+   +---+   +
|               |
+   +---+   +   +
|   |       |   |
+---+   +---+---+
|               |
+---+---+---+---+
```

Assignment

- Due
  - **~ 2023.04.26(Wed) 23:59**
  - Last Commit 기준

- 자세한 내용은 과제 명세 PDF 파일 참고