SUDOKU & CROSSWORD

SIXA

(이소령 이창우 이한별 신은영 이지연 임솔빈)

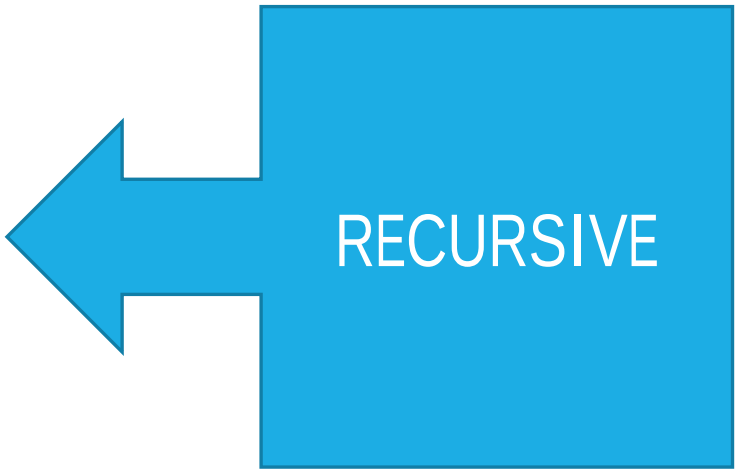
INDEX

GOAL

Donald Knuth의 Dancing Links 알고리즘을 이용하여 Sudoku 퍼즐 문제를 해결하고 더 나아가 크로스워드 퍼즐 문제를 해결하는 알고리즘을 고안한다.

SUDOKU

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



```
int fillSudoku(int puzzle[][9], int row, int col)
{
    int i;
    if (row<9 && col<9)
    {
        if (puzzle[row][col] != 0)
        {
            if ((col + 1)<9) return fillSudoku(puzzle, row, col + 1);
            else if ((row + 1)<9) return fillSudoku(puzzle, row + 1, 0);
            else return 1;
        }
        else
        {
            for (i = 0; i<9; ++i)
            {
                if (isAvailable(puzzle, row, col, i + 1))
                {
                    puzzle[row][col] = i + 1;
                    if ((col + 1)<9)
                    {
                        if (fillSudoku(puzzle, row, col + 1)) return 1;
                        else puzzle[row][col] = 0;
                    }
                    else if ((row + 1)<9)
                    {
                        if (fillSudoku(puzzle, row + 1, 0)) return 1;
                        else puzzle[row][col] = 0;
                    }
                    else return 1;
                }
            }
        }
        return 0;
    }
    else return 1;
}
```

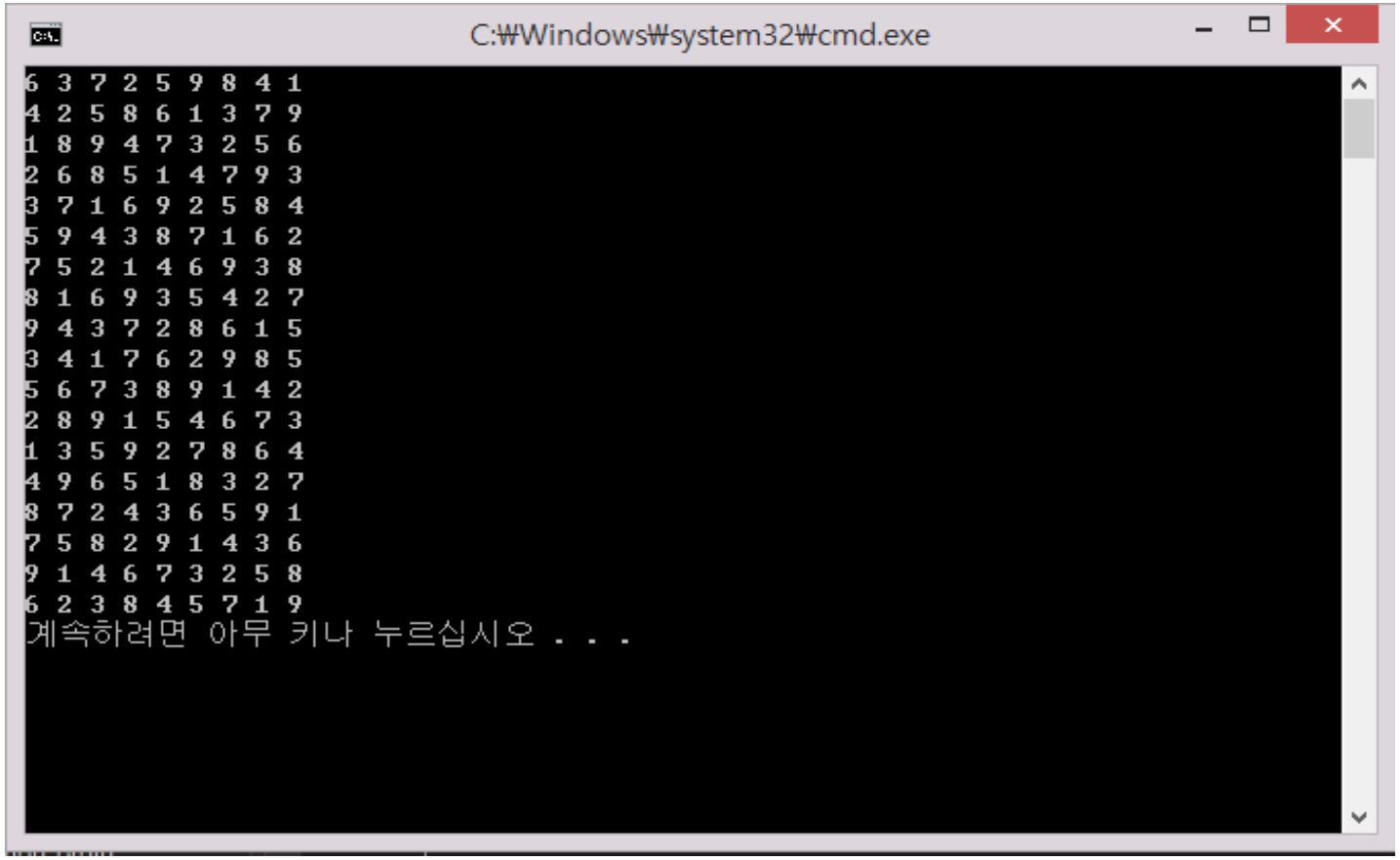
SUDOKU : Recursive Algorithm

```
int fillSudoku(int puzzle[][9], int row, int col)
{
    int i;
    if (row<9 && col<9)
    {
        if (puzzle[row][col] != 0)
        {
            if ((col + 1)<9) return fillSudoku(puzzle, row, col + 1);
            else if ((row + 1)<9) return fillSudoku(puzzle, row + 1, 0);
            else return 1;
        }
        else
        {
            for (i = 0; i<9; ++i)
            {
                if (isAvailable(puzzle, row, col, i + 1))
                {
                    puzzle[row][col] = i + 1;
                    if ((col + 1)<9)
                    {
                        if (fillSudoku(puzzle, row, col + 1)) return 1;
                        else puzzle[row][col] = 0;
                    }
                    else if ((row + 1)<9)
                    {
                        if (fillSudoku(puzzle, row + 1, 0)) return 1;
                        else puzzle[row][col] = 0;
                    }
                    else return 1;
                }
            }
        }
        return 0;
    }
    else return 1;
}
```

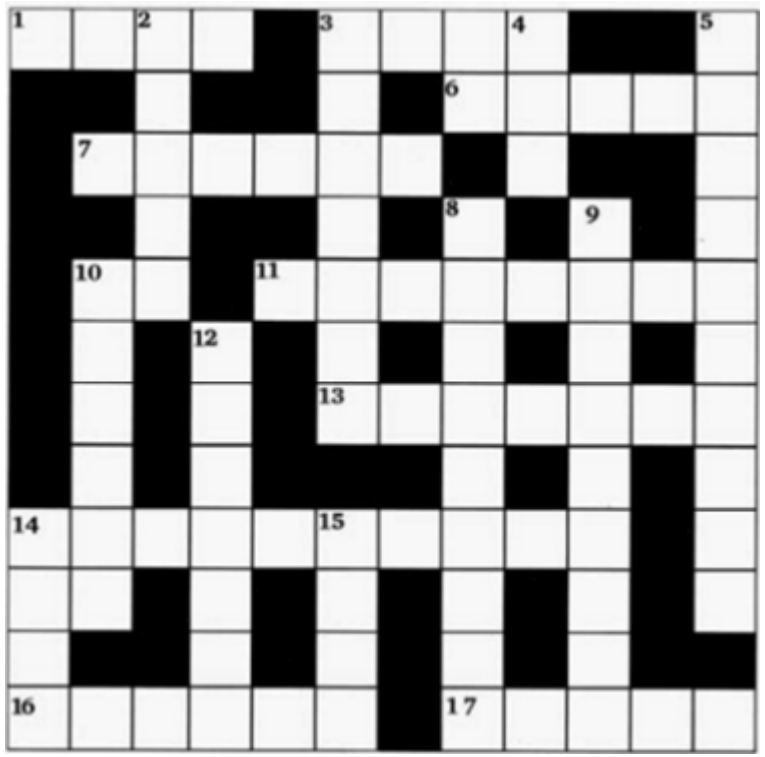
Recursive of fillSudoku()

: 다음 col과 row를 확인해서 부합하는 숫자일 경우 Return 1

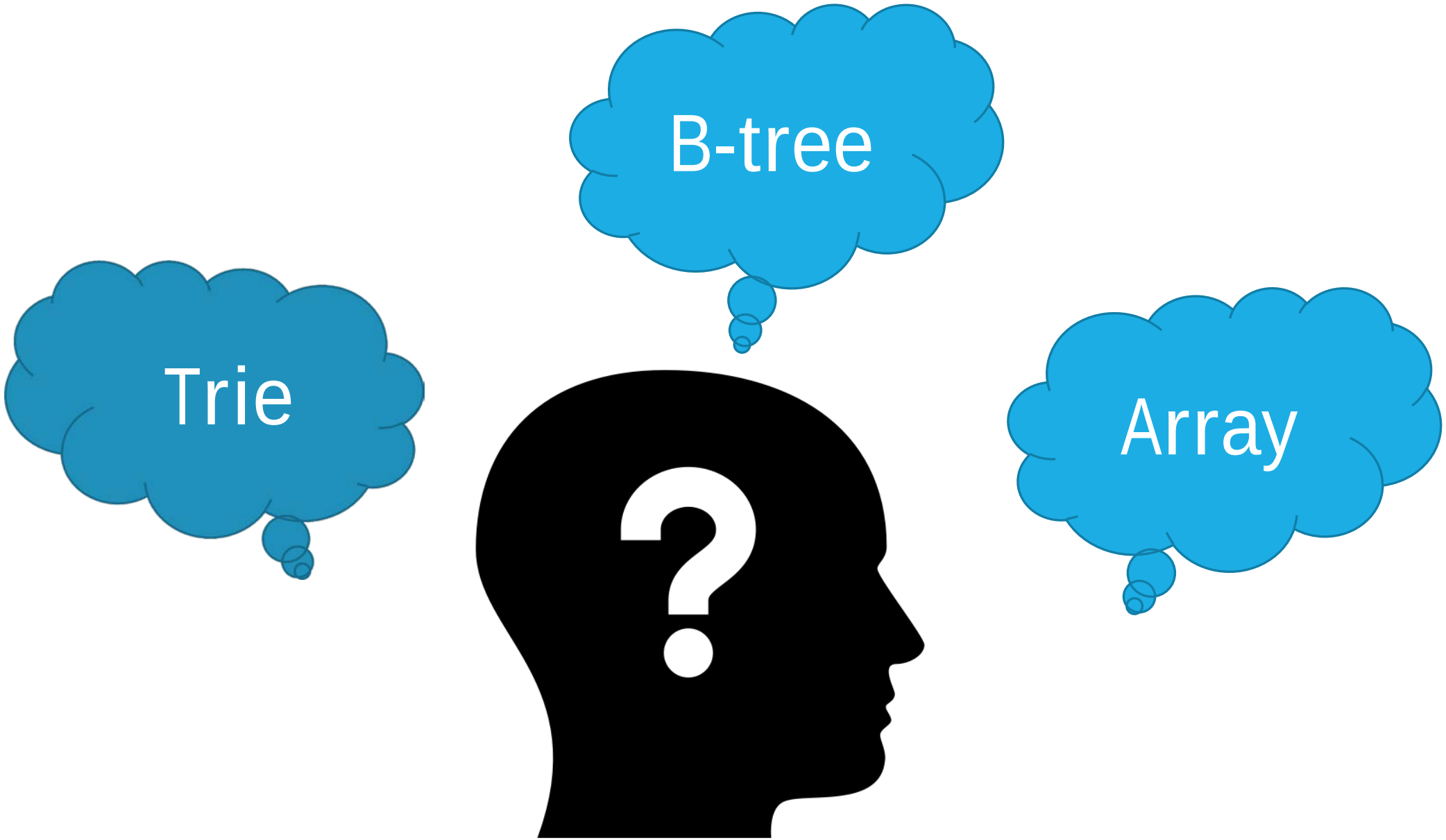
SUDOKU : Print Output



CROSSWORD PUZZLE



DANCINGLINK
+
Data
structure


B-tree
Trie
Array

CROSSWORD PUZZLE : Functions

바뀐 신호를 받으면 list와 사전 포인터 갱신

현재 워드 _칸 조사

교차점 조사

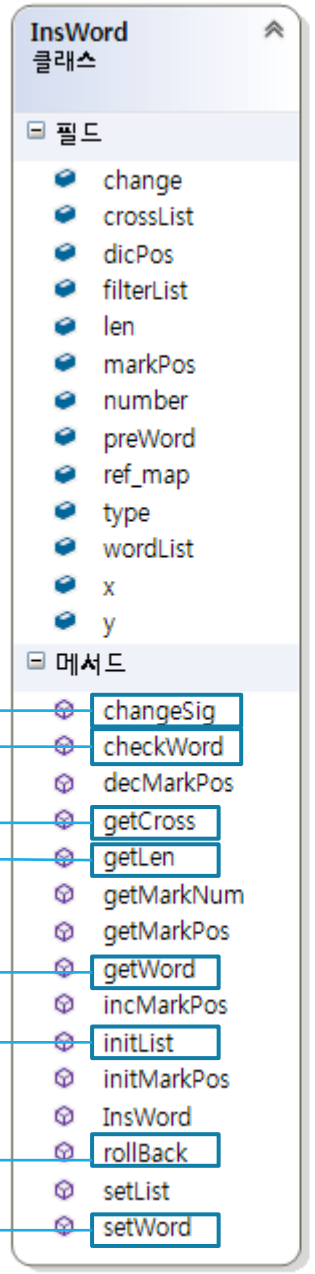
현재 객체 길이

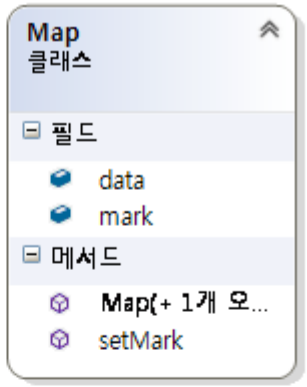
현재 객체 단어

단어 사전 리스트 초기화

기입한 단어 복구

map에 단어 기입





THANK YOU