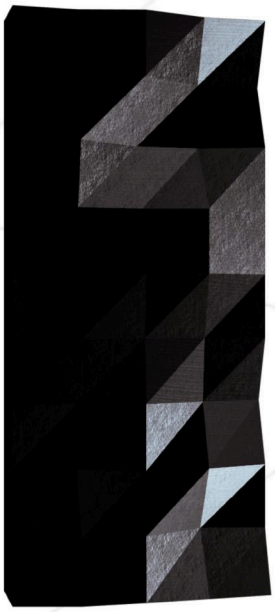


Dash – Problem solving_02

Summary: this document is the subject for the dash @ 42Seoul.

Contents

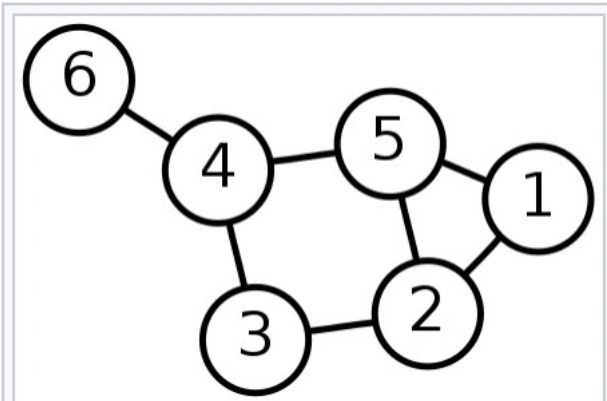
Chapter 1

Foreword

This project focuses on solving problems and aims to develop a diverse perspective on problems.

Chapter 2

Objective



A drawing of a graph.

https://en.wikipedia.org/wiki/Graph_theory

I recommend that you study graph theory and proceed.

In Graph Theory, Depth First Search (DFS) and **BFS (Bread First Search) account for many parts. In fact, it is a technology that is used as a base for navigation in many places, such as finding alignment (Quick, Merge) and shortest distance (Navigation).

Chapter 3

Instructions

- Include -Wall -Wextra -Werror for build options.
- I recommend using global variables
- There are limitations for each problem, so please read RedBox carefully.
- We don't keep norm.
- <stdio.h> is available.

Chapter 4

Exercise 00 : villages

	Exercise 00
villages	
Turn-in directory : ex00/	
Files to turn in : villages.c	
Allowed function : write	

N * N maps are present. One is where the house is, and zero is where the house is.

Houses connected by east, west, south, and north can be talked about as a village, write a program to find the number of villages.

3 <= N <= 50

input:

6

111111

100100

011111

011001

100111

101110

output:

2

input:

7

0111100

0101000

0010110

0100111

0101000

1000010

1101110

output:

7

Chapter 5

Exercise 01 : Go home

	Exercise 01
Go home	
Turn-in directory : ex01/	
Files to turn in : go_home.c	
Allowed function : write	

Exercise 01:

ssj of "Sgang", "Seungyel", and "Johokim" could not pass through "exam05" and decided to enter the maze and conduct a closed tube training. After finishing the closed coffin training, Seung-yel tried to leave the maze, but his memory was poor, so he forgot the exit of the maze.

Please let Seungyel escape the maze before he dies.

N * M maze is given. (2 <= N, M <= 50)

Find the shortest path to the coordinates of [N - 1, M - 1], which is located at [0, 0]..

input:	input:
5 5	5 7
11111	1111111
10001	1010101
10001	1010111
10001	1010111
11111	1111101
output:	output:
9	11