



RO:BIT
ROBOT SPORT GAME TEAM

Team. RO:BIT | C++ & qt day3

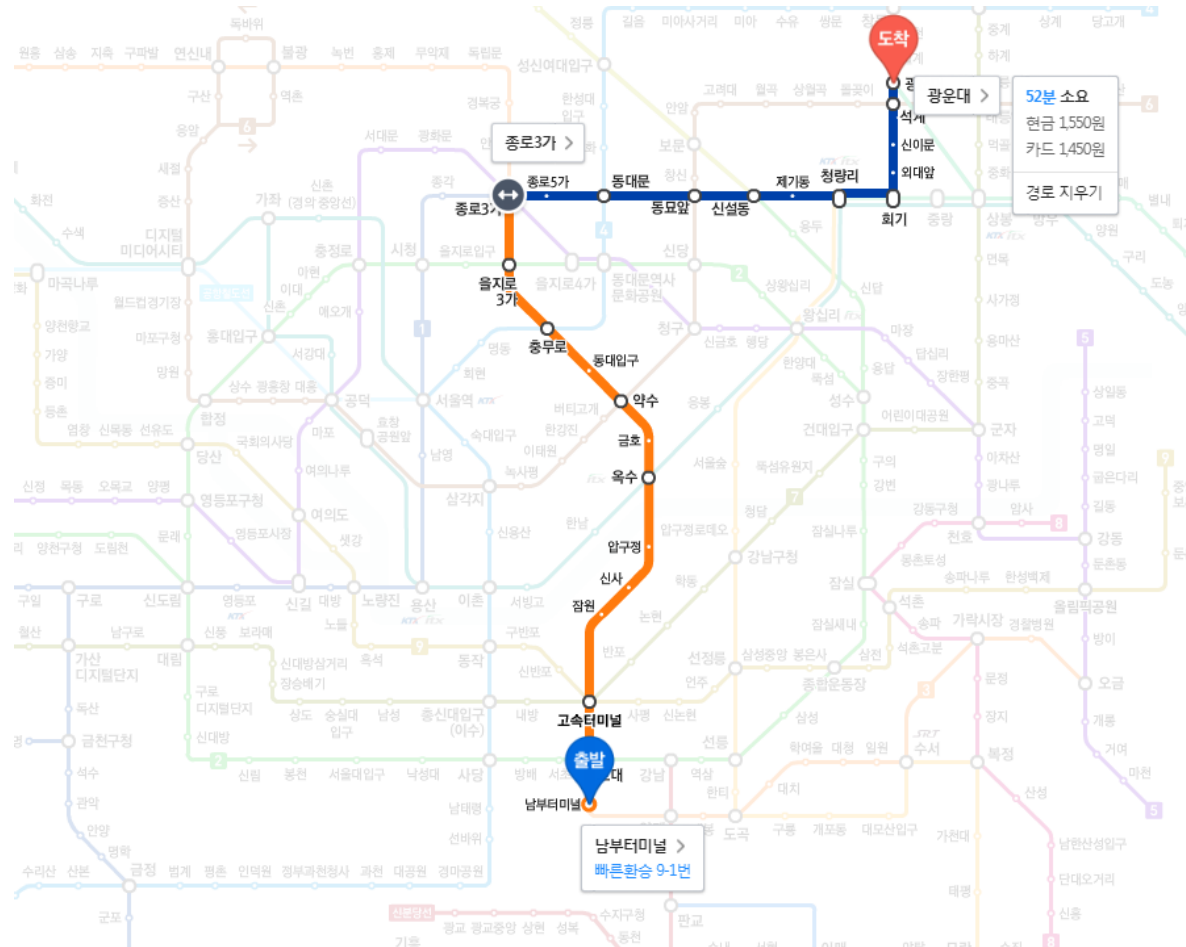
18기 이선경

알고리즘

- 어떤 작업을 수행하기 위해 입력을 받아
- 원하는 출력을 만들어 내는 과정을 기술한 것
- 입력을 출력으로 변환하는 계산 절차

해야 할 작업을 명확하게 기술
→ 입력과 출력을 명확하게 기술

입력: 지하철 노선도, 출발역,
도착역
출력: 출발역에서 도착역까지
가는 최단 경로



알고리즘

- 어떤 작업을 수행하기 위해 입력을 받아
- 원하는 출력을 만들어 내는 과정을 기술한 것
- 입력을 출력으로 변환하는 계산 절차

해야 할 작업을 명확하게 기술
→ 입력과 출력을 명확하게 기술

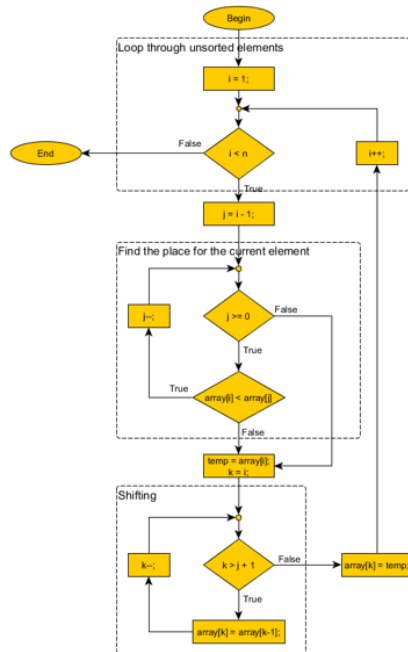
입력: 고객 n 명의 위치, 출발
위치, 지도
출력: 고객들을 모두 방문하고
돌아오는 최단 경로



알고리즘

- 명확하게
 - 모호하지 않게
 - 필요한 수준까지 구체화
 - 특정 프로그래밍 언어로 변환하는데 어렵지 않을 정도

순서도



의사 코드

INSERTION-SORT (A)

```
for j = 2 to A.length
    key = A[j]
    i = j - 1
    while i > 0 and A[i] > key
        A[i+1] = A[i]
        i = i - 1
    A[i+1] = key
```

알고리즘

소요 시간 (시간 복잡도)
메모리 (공간 복잡도)

관심의 대상: 입력의 크기가 충분히 클 때
→ 점근적 분석

| 알고리즘

정렬

[선택/버블/삽입/병합/퀵/셸/힙 등]

탐색

[행렬, 배열, 트리, 그래프]

트리

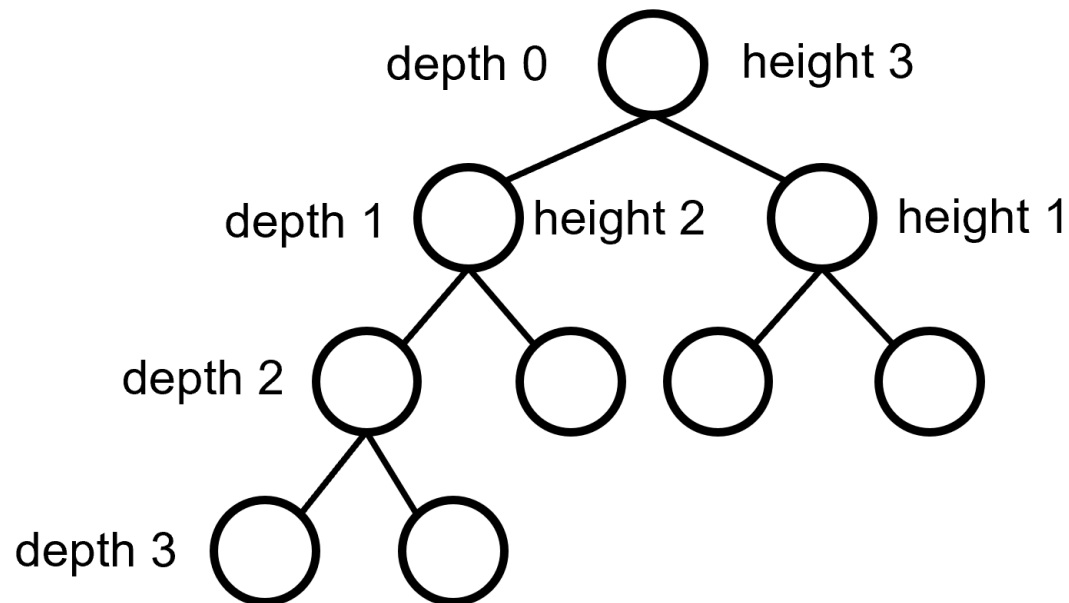
[이진탐색 트리, AVL, 스펠레이, 레드블랙 트리, B트리, KD트리]

그래프

[BFS, DFS, 프림, 크루스칼, 다익스트라, 벨만-포드, A*]

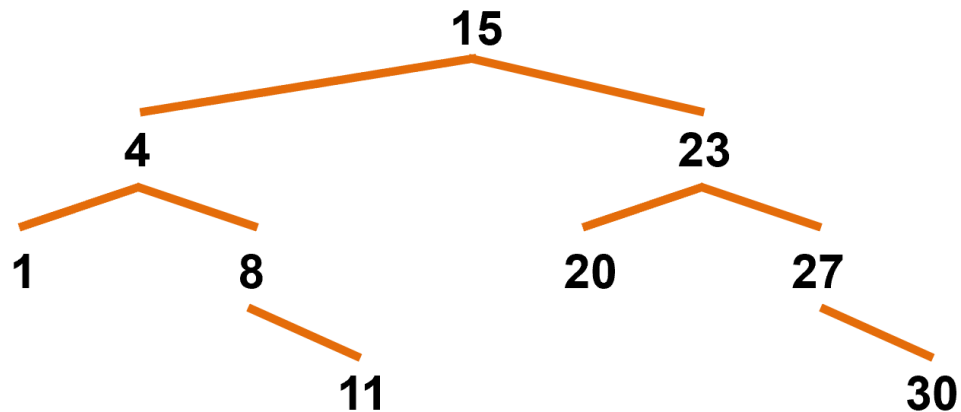
알고리즘-트리

- 노드의 깊이(depth): 루트 노드에서 해당 노드까지의 경로에 있는 간선(edge)의 개수
- 노드의 높이(height): 해당 노드에서 리프 노드까지의 가장 긴 경로에 있는 간선(edge)의 개수
- 트리의 높이: 루트 노드의 높이



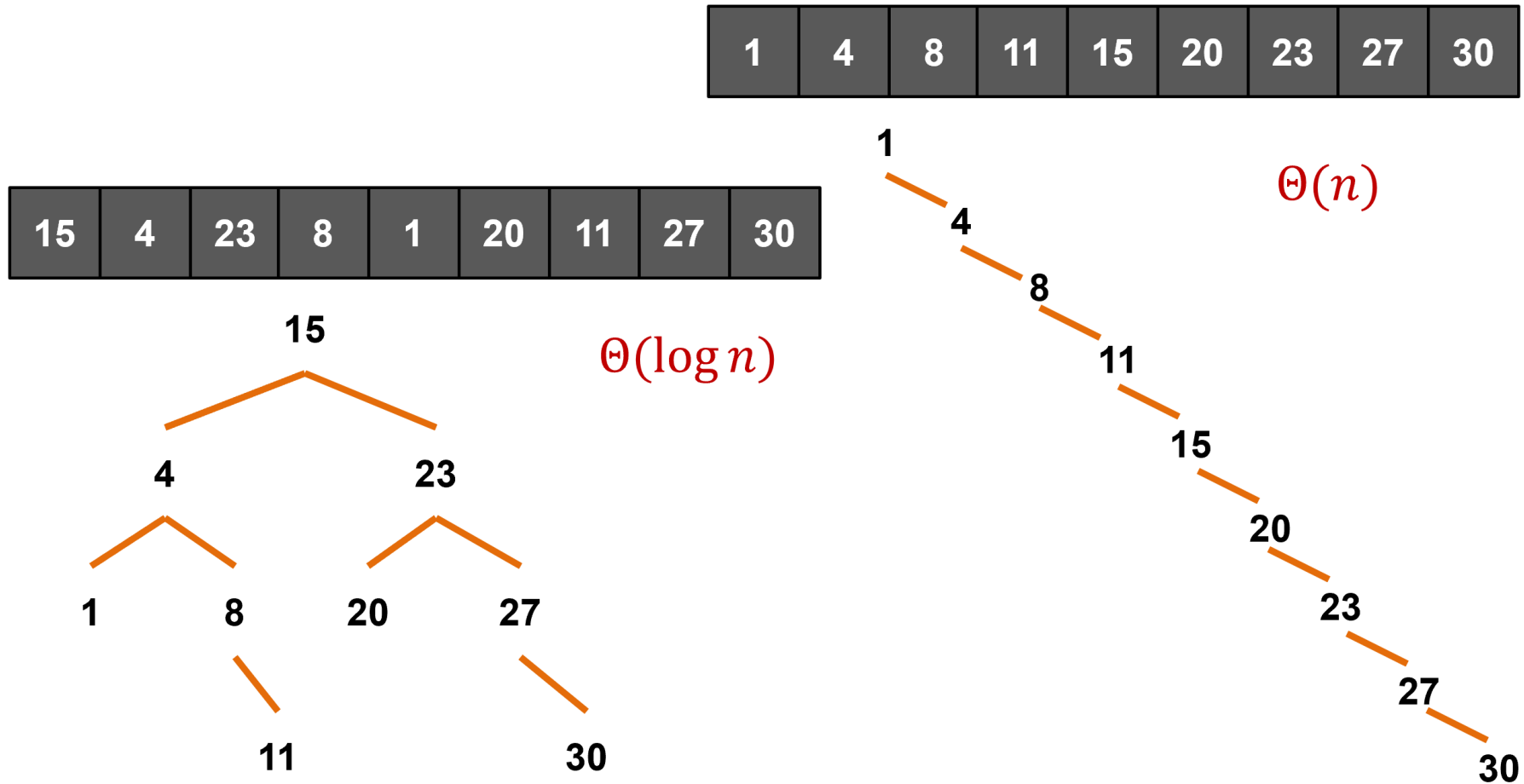
알고리즘-트리

- 임의의 데이터들
→ 순서대로 정렬한 후 이진 탐색
- 데이터를 입력할 때부터 정렬
→ 이진 탐색(검색) 트리

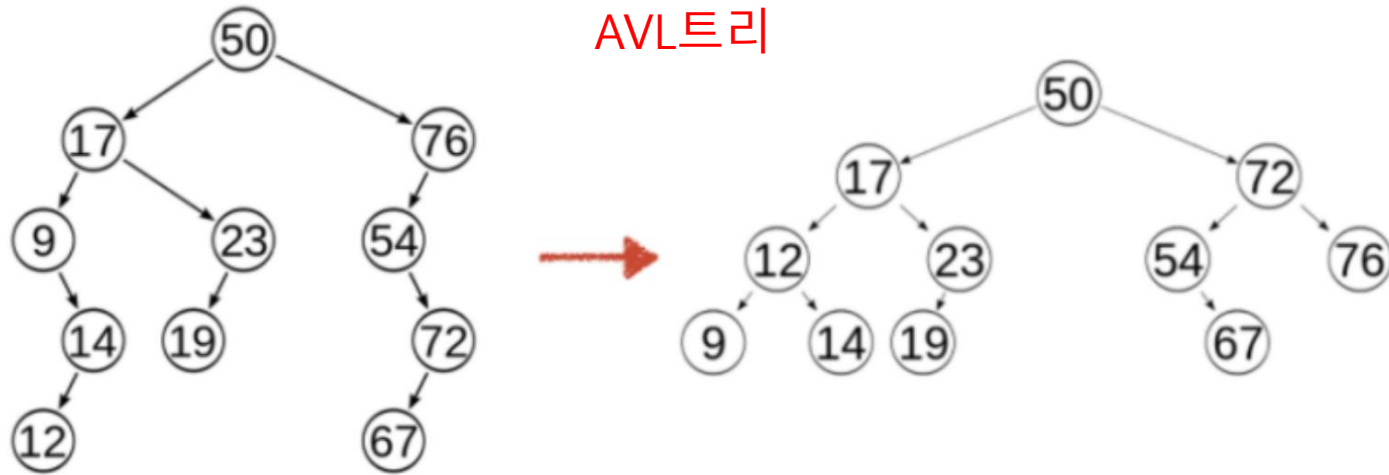


알고리즘-트리

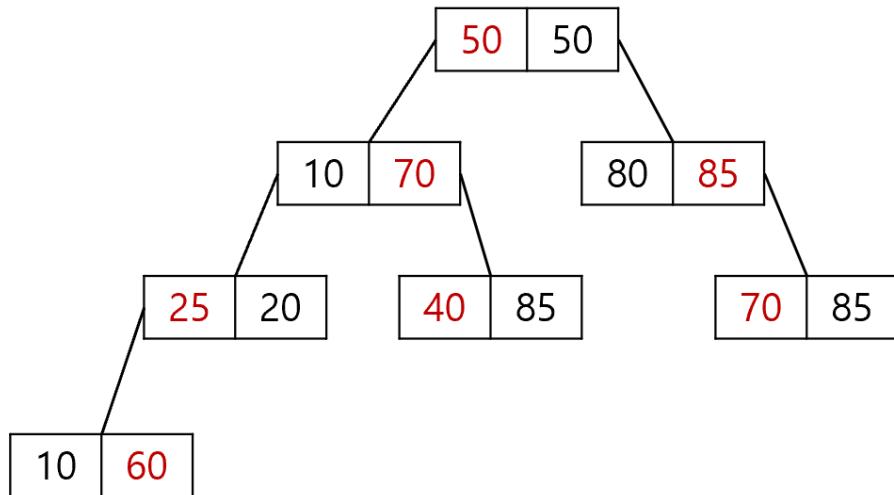
- 데이터 입력 순서에 따라 성능이 달라진다.



알고리즘-트리

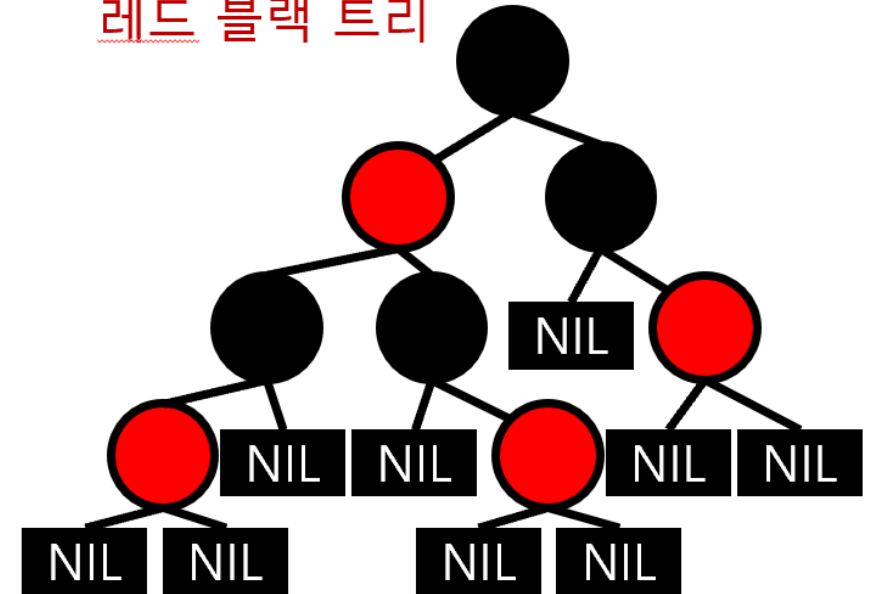


- (50,50), (10,70), (80,85), (25,20), (40,85), (70,85), (10,60)



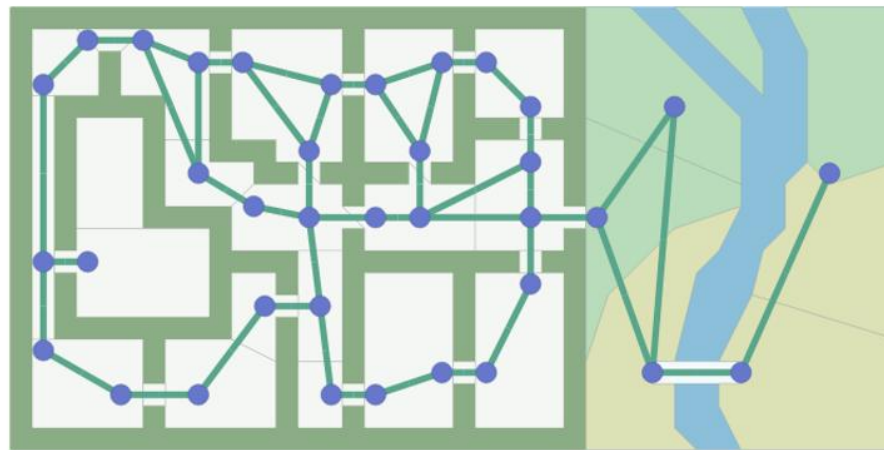
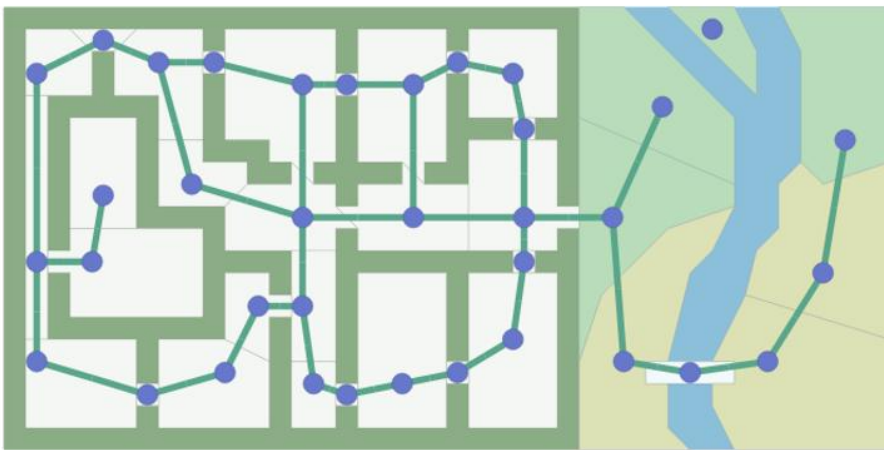
KD-트리

레드 블랙 트리

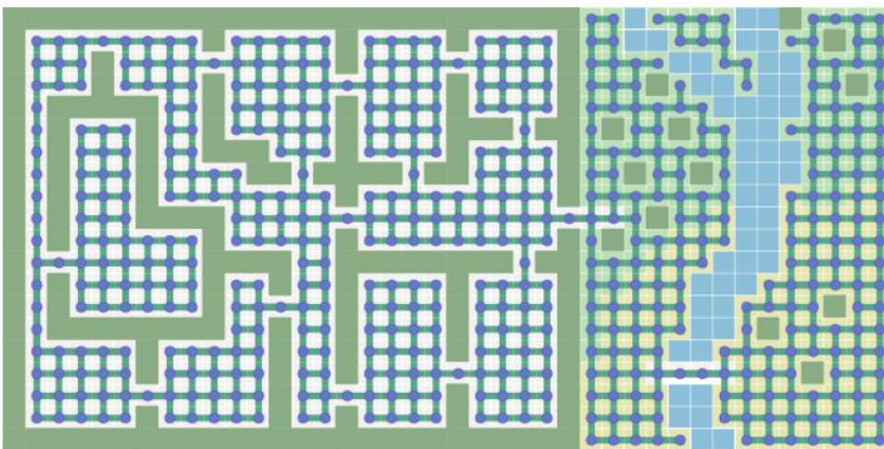


알고리즘-그래프

목적에 따라 여러 형태로 표현 가능

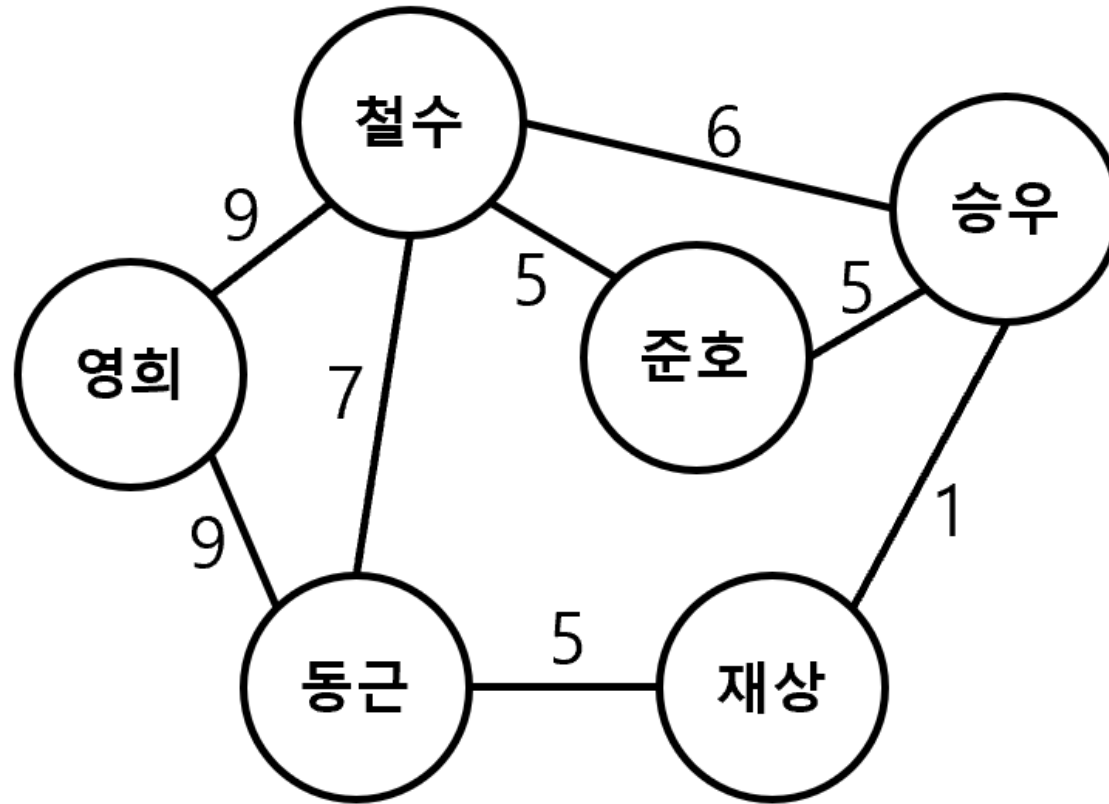


www.redblobgames.com



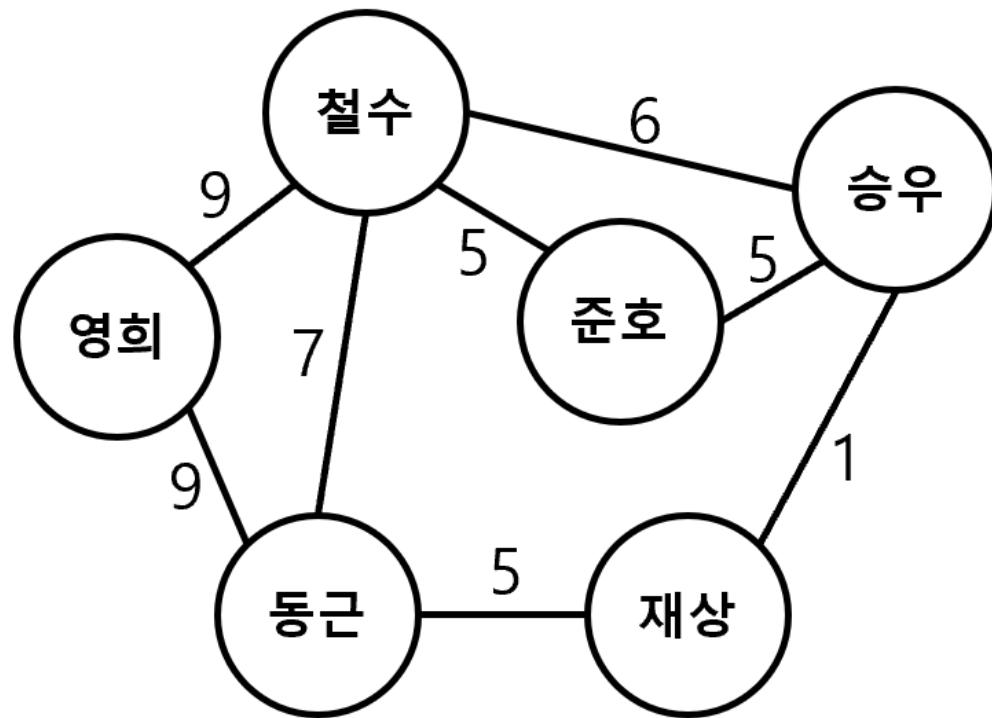
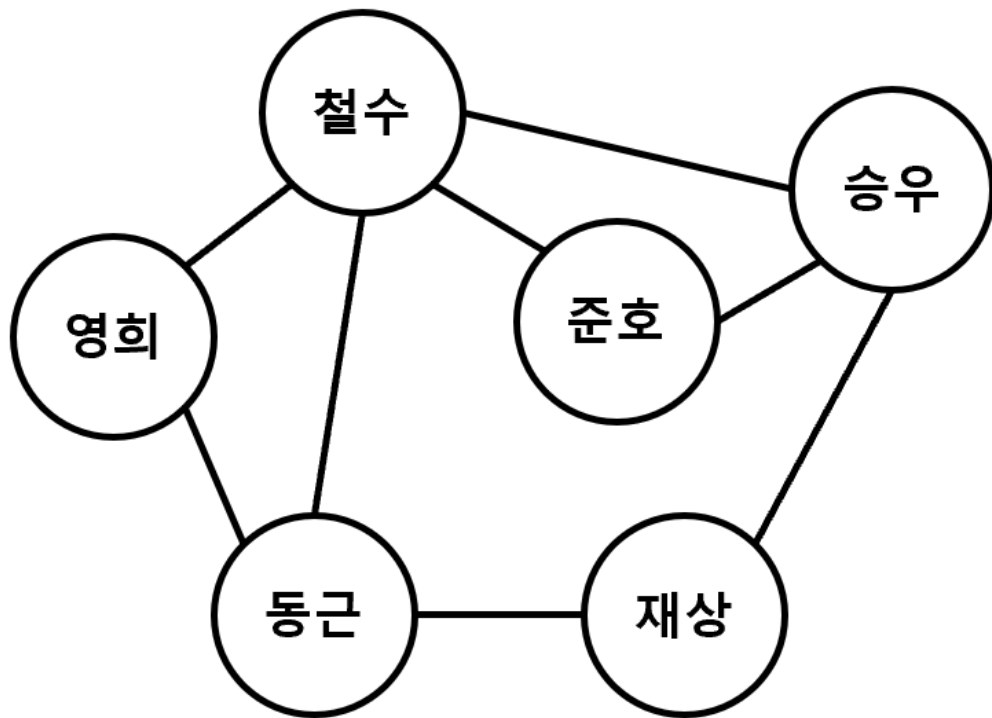
알고리즘-그래프

- 정점 (Vertex)
- 간선 (Edge)
- 가중치 (Weight)



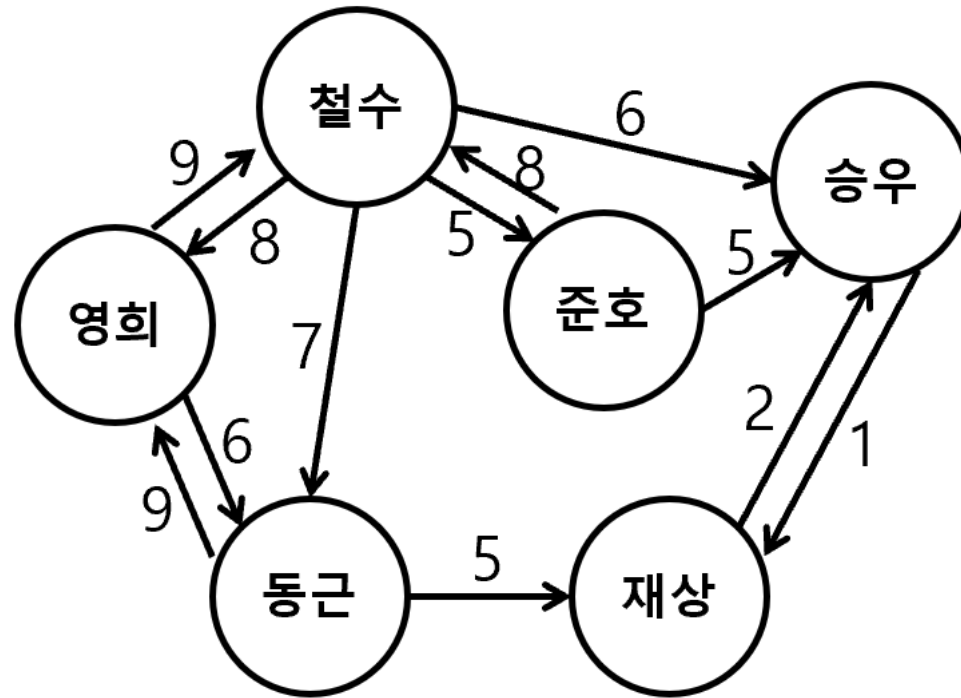
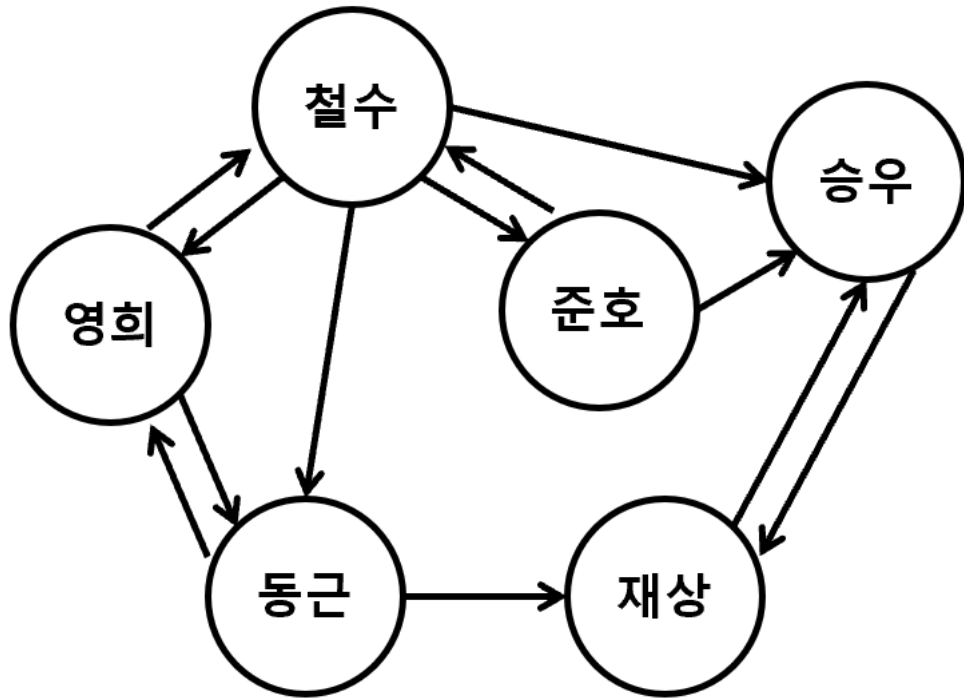
알고리즘-그래프

- 무향 그래프 (Undirected Graph)



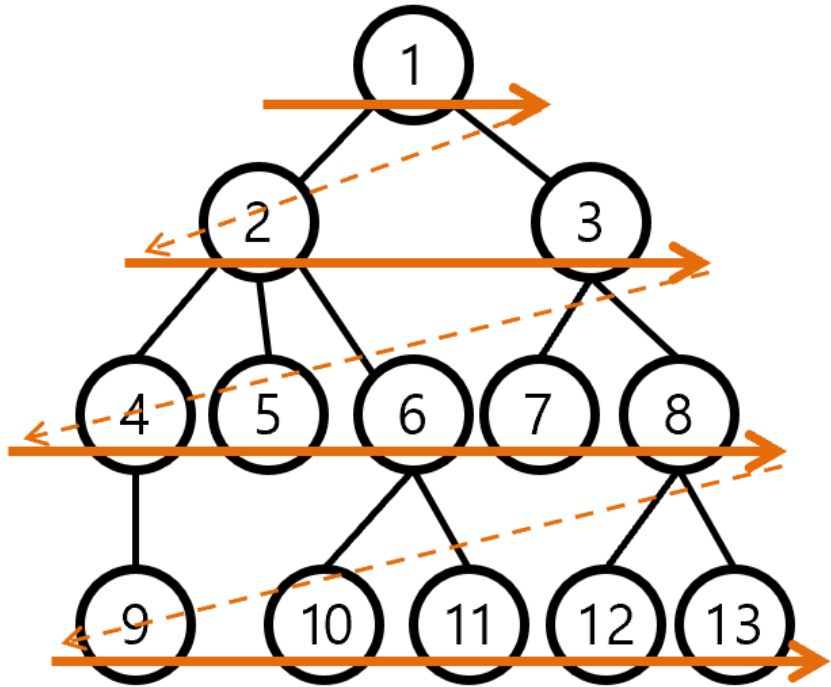
알고리즘-그래프

- 유향 그래프 (Directed Graph)

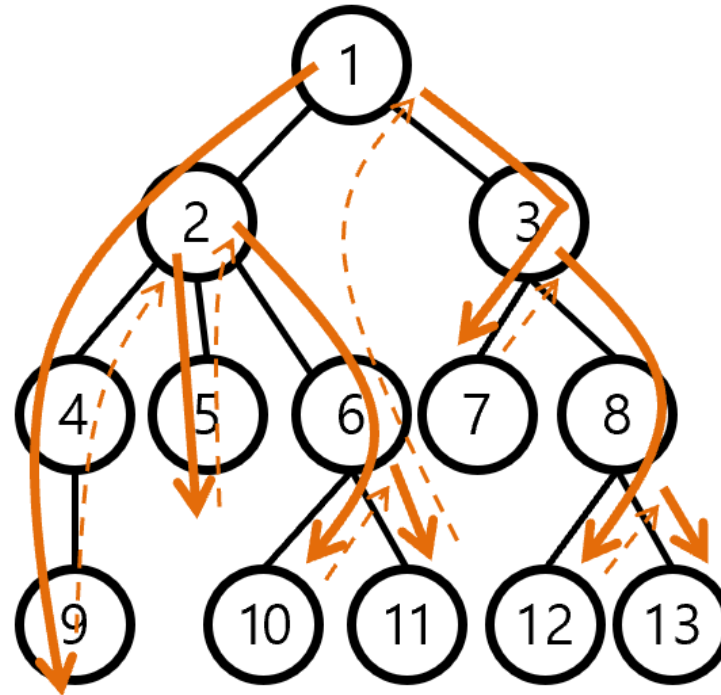


알고리즘-그래프

- 너비 우선 탐색
(Breadth-First Search; BFS)



- 깊이 우선 탐색
(Depth-First Search; DFS)



알고리즘-최단경로탐색(Dijkstra)

지도 데이터 (벽: 0, 길: 비용 표시)

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0
0	3	0	0	0	0	2	0	1	0
0	3	1	1	1	0	1	0	1	0
0	0	1	0	1	0	1	1	1	0
0	1	1	0	1	1	1	0	1	0
0	0	0	0	0	0	0	0	0	0

각 칸을 구별하기 위한 인덱스

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	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70

경로 기록

12 → 13
 52 → 53
 53 → 43
 43 → 33
 33 → 34
 33 → 32
 34 → 35
 35 → 45
 32 → 22
 45 → 55
 55 → 56
 56 → 57
 22 → 12
 57 → 47
 12 → 13
 47 → 48
 47 → 37
 13 → 14
 48 → 49
 37 → 27
 14 → 15
 49 → 39
 49 → 59
 27 → 17
 15 → 16
 39 → 29
 17 → 18
29 → 19

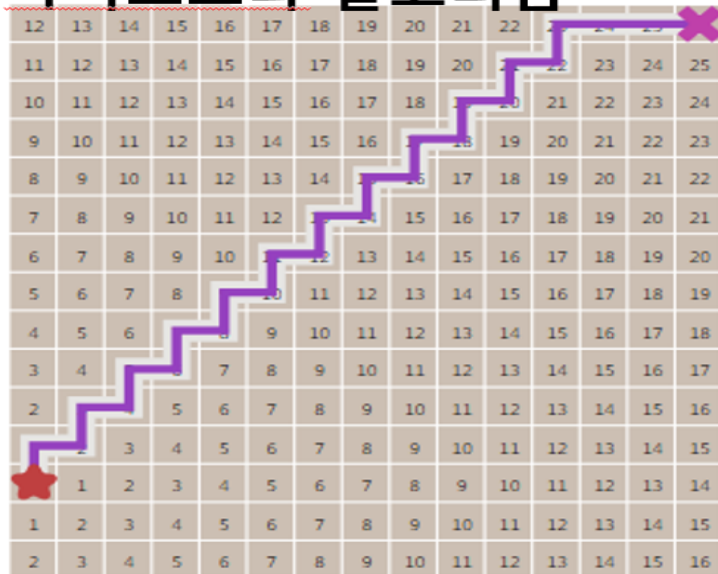
누적 비용

0	0	0	0	0	0	0	0
0					0		0
0	0	0	0		0		0
	0		0		0	0	0
0	0		0	0	0		0

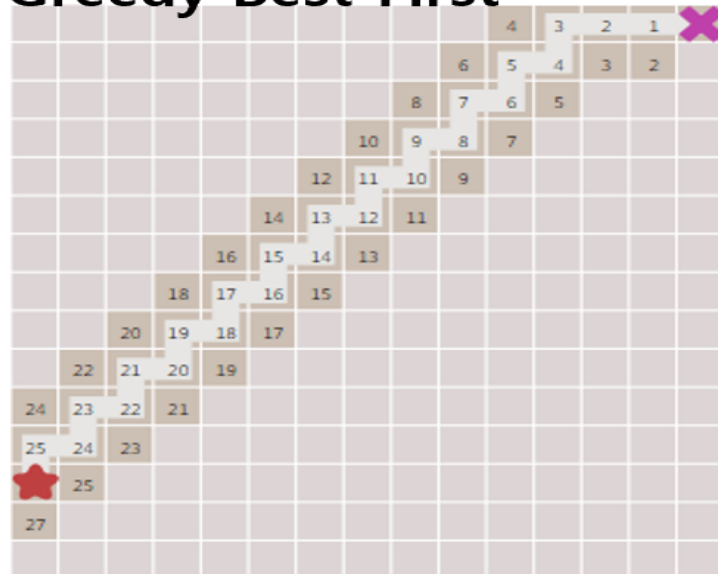
10	11	12	13	14	14	15	15
9					13		14
6	3	4	5		11		13
	2		6		10	11	12
0	1		7	8	9		13

알고리즘-그래프

다익스트라 알고리즘



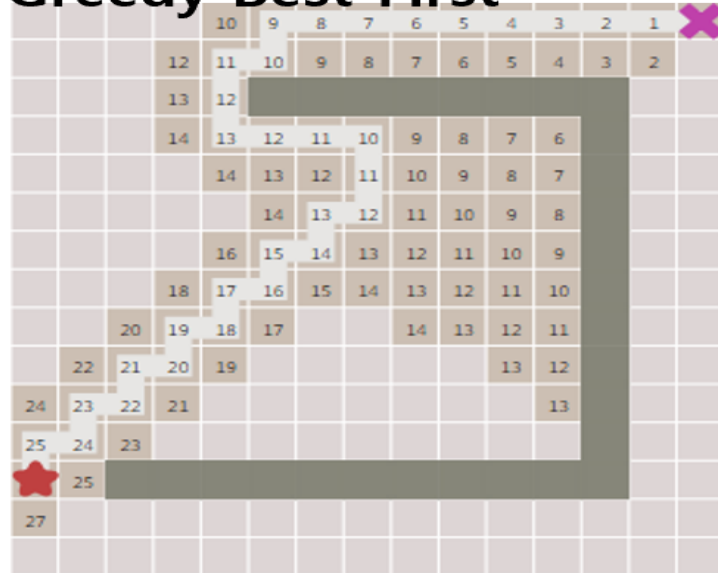
Greedy Best-First



다익스트라 알고리즘



Greedy Best-First



과제1

천지인 키보드 만들기

아래 사진의 키보드와 같이 구현



과제1

천지인 키보드 과제

[필수조건]

- 대소문자 구현
- 입력창에 실시간으로 작성중인 문자 출력
- 엔터가 눌리면 문장 txt파일 마지막줄에 추가

[추가점수]

- 한글자판 구현
- 숫자 및 특수문자 구현
- 입력 받은 버튼의 눌림 처리(배경색상 변화 등)
- 기타 실제 천지인 키보드와 유사할수록 추가점 부여

* mp4파일로 문장 출력 결과영상 첨부할것

| 과제2

알고리즘 보고서 작성

11~15pt,
목차, 표지, 인사말 **제외** 최소 20p 이상

알아보기 쉽도록 개조식으로 작성

과제3

A* 알고리즘 , dijkstra 알고리즘 구현 및 비교

참고 영상 링크

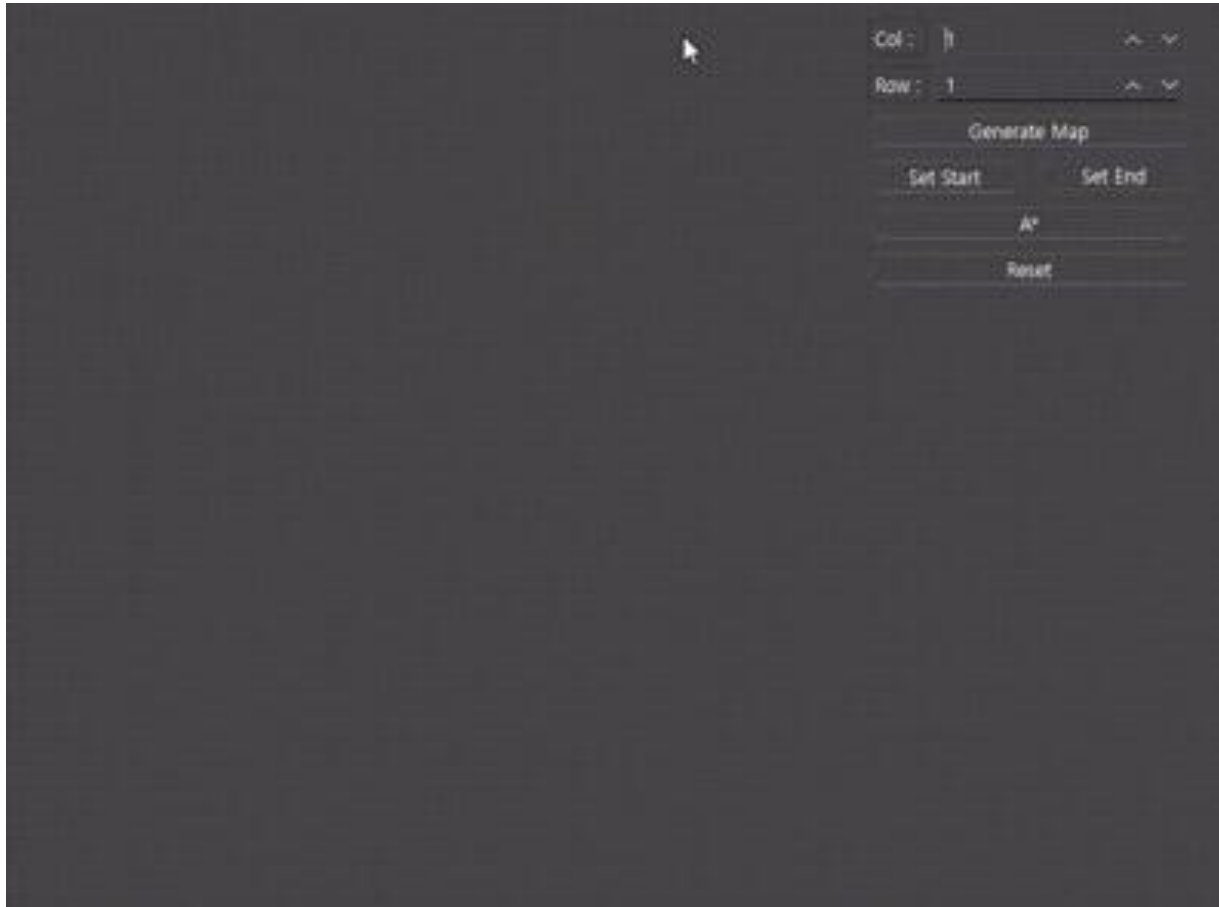
<https://www.youtube.com/watch?v=g024lzsknDo>

필수 구현 기능:

- 맵 사이즈 구현
- 수동 장애물, 시작점, 도착점 설정
- 탐색 과정 시각화

과제 3

참고자료) Astar qt 구현 예시



결과영상 mp4 파일로 제출
구현 보고서 작성

| 과제3

GIT 제출

제출 commit 시간 기준 채점, 월요일 출근(7시) 전까지

제출기한 및 양식 미준수 시 채점안함

Thank you
