



자료구조 팀 프로젝트 최종 발표

미로 찾기

Team 9

박상희
김제현
정주원
임수연



Team

Presentation Index

1. 미로 찾기 분석
 - 정의, 미로 성질 등
2. 탐색 알고리즘 조사 및 분석
 - Uninformed & Informed Search
3. 탐색 알고리즘 구현 결과 확인
 - DFS, BFS, 다익스트라, A*



미로 찾기 분석



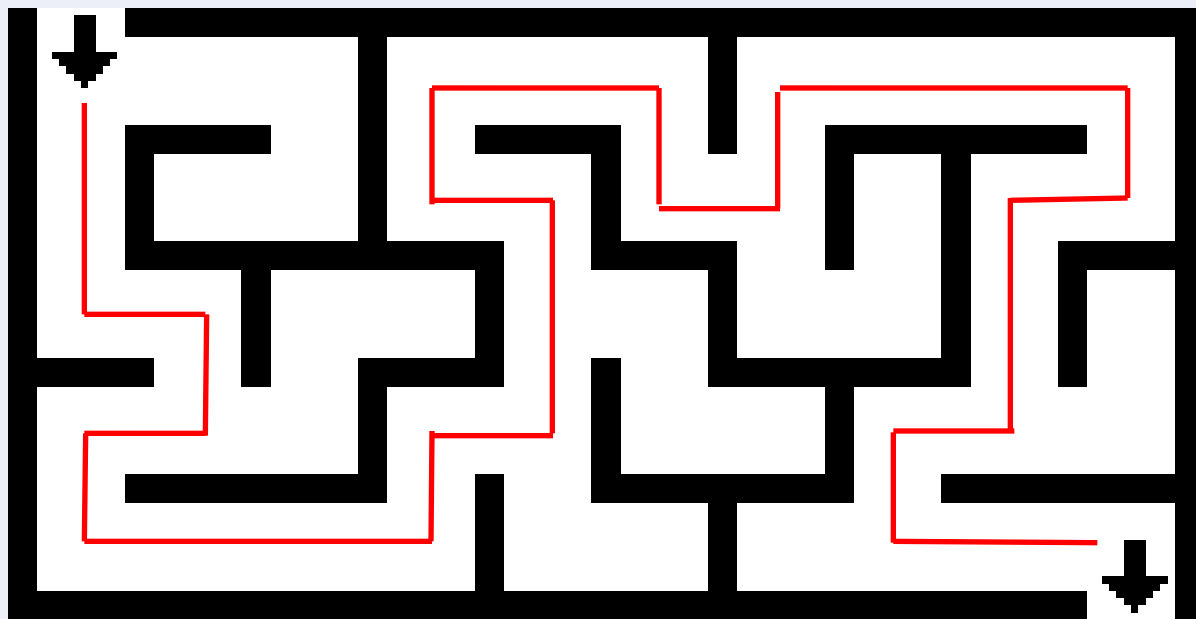
팀 프로젝트 '미로 찾기'

미로 찾기 분석

- 미로 찾기란

미로 : 복잡한 길을 찾아 출발점부터 시작해 도착점까지 도달하는 퍼즐

미로 찾기 : **출발점에서 도착점으로 가는 경로를 찾는 것**





팀 프로젝트 '미로 찾기'

미로 찾기 분석

- 미로의 성질

이동	실제로 이동	가상으로 이동
정보	Uninformed	Informed
순환구조	있음	없음
도착지	있음	없음
크기	유한	무한
가중치	있음 [=불균일]	없음 [=균일]

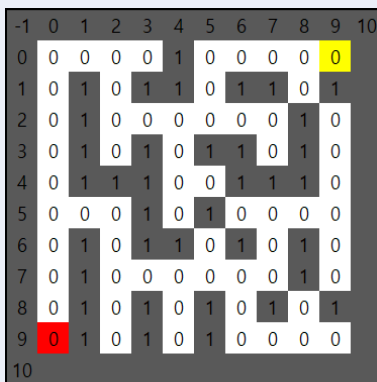


팀 프로젝트 '미로 찾기'

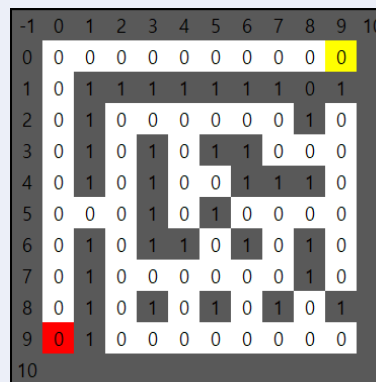
미로 찾기 분석

- 미로의 유형

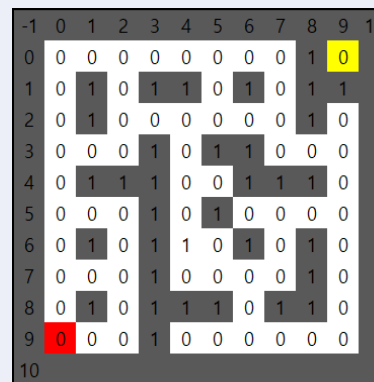
Case1. 순환x



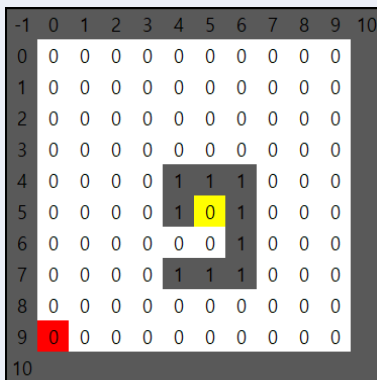
Case2. 순환o



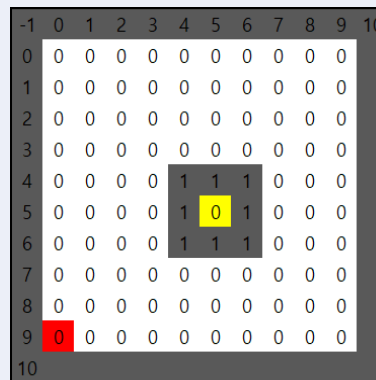
Case3. 순환o, 경로x



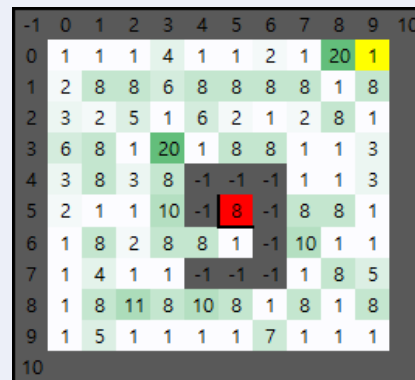
Case4. 공터



Case5. 공터, 경로x



Case6. 공터, 가중치0





팀 프로젝트 '미로 찾기'

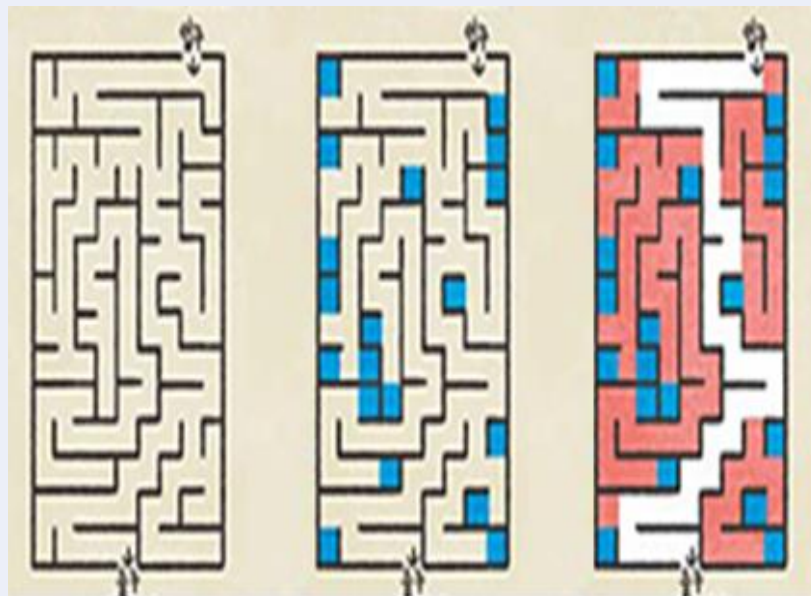
미로 찾기 분석

- 단순하게 미로를 푸는 방법

1) 벽을 따라서 탐색하는 방법



2) 막다른 공간을 칠하는 방법





탐색 알고리즘 조사 및 분석



팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

도메인에 대한 정보 없이 목적지 노드를 찾는 것



$$f(n) = g(n)$$



팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

알고리즘 평가요소

- Completeness: 답이 존재하면 항상 찾을 수 있는가?
- Optimality : 찾아낸 답이 최소 비용인가? 더 좋은 답은 없는가?
- Time Complexity : 시간 복잡도
- Space Complexity : 공간 복잡도

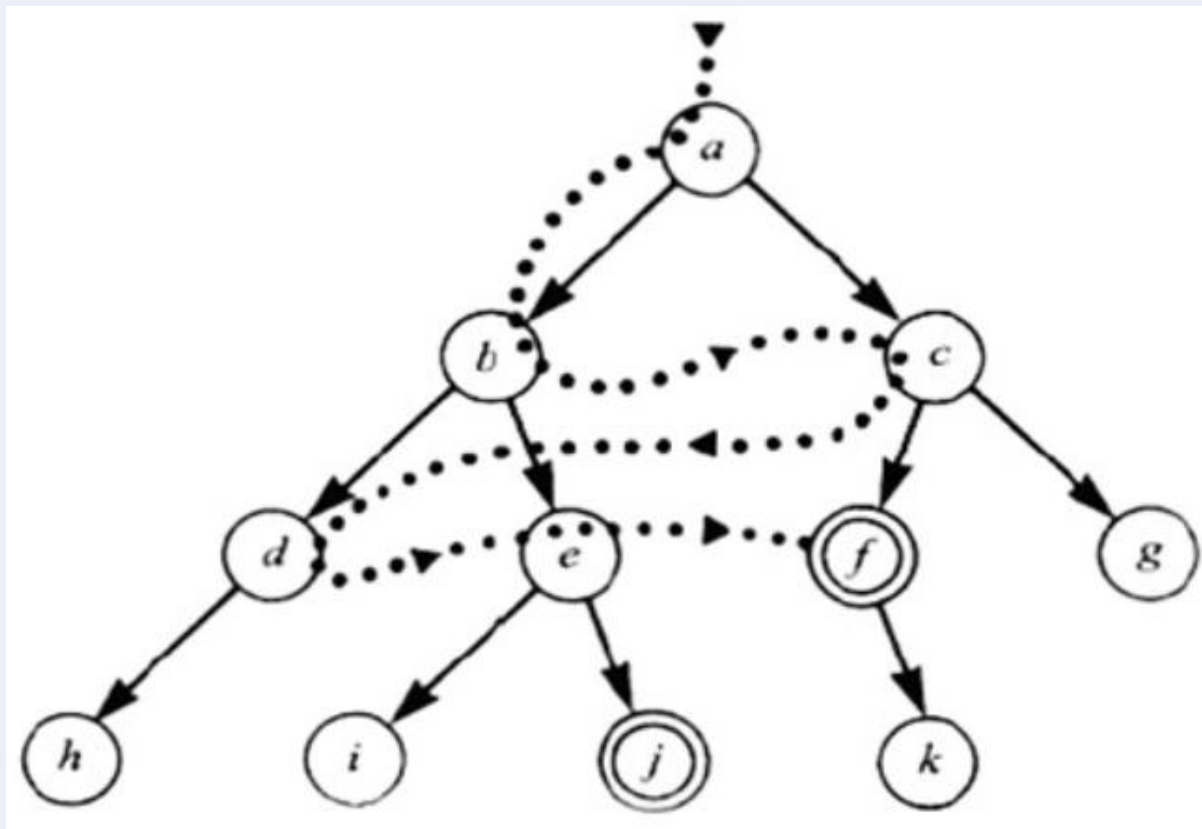


팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

BFS(Breadth First Search, 너비 우선 탐색)



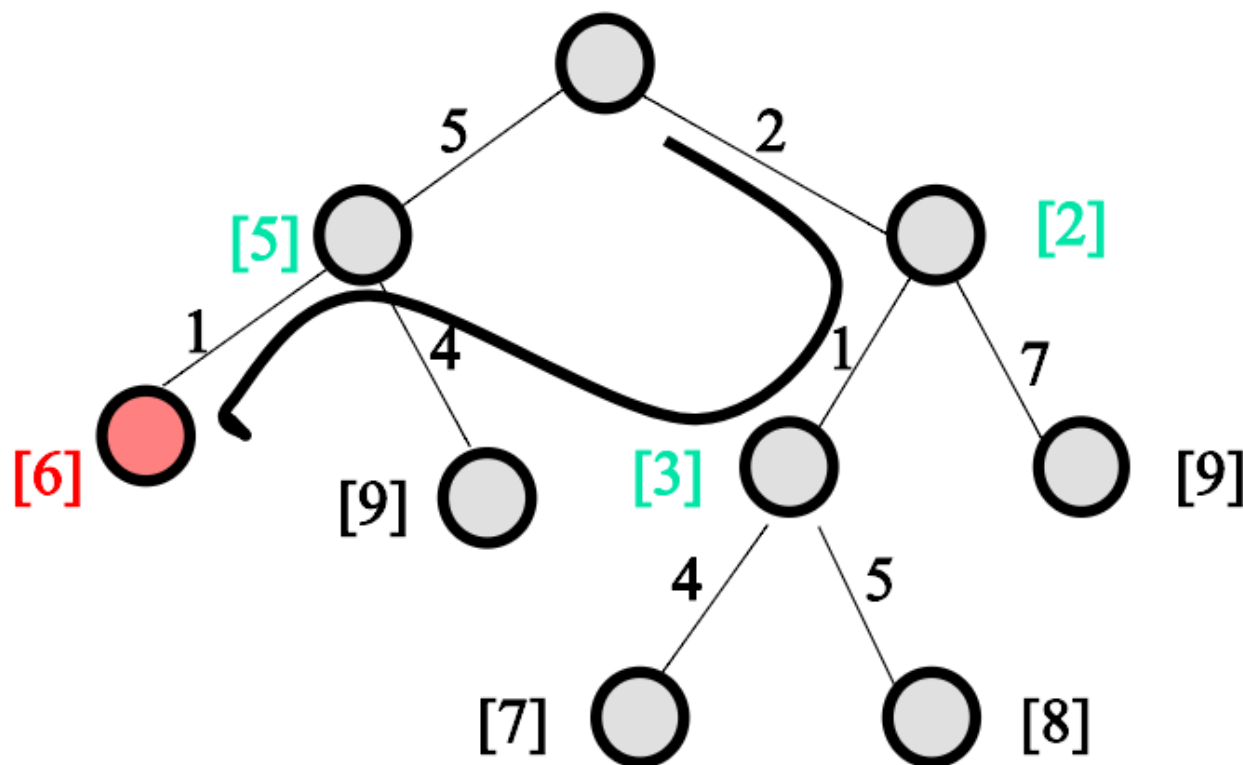


팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

Uniform-cost Search, Lowest-cost Search



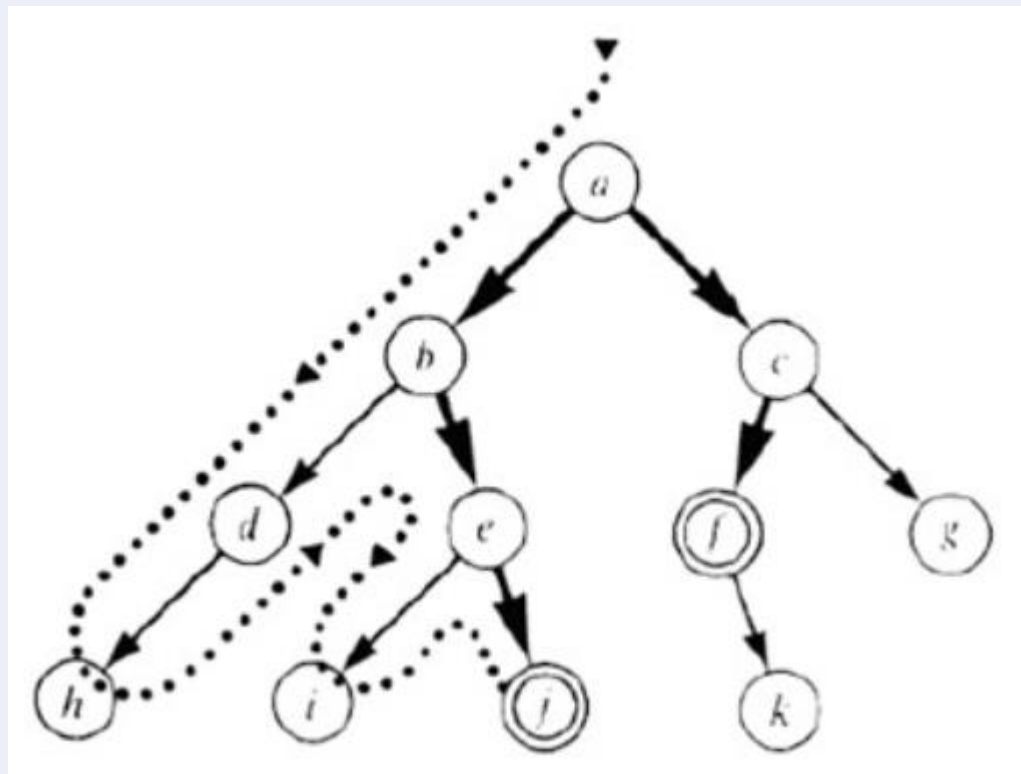


팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

DFS(Depth First Search, 깊이 우선 탐색)





팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

Depth-limited Search



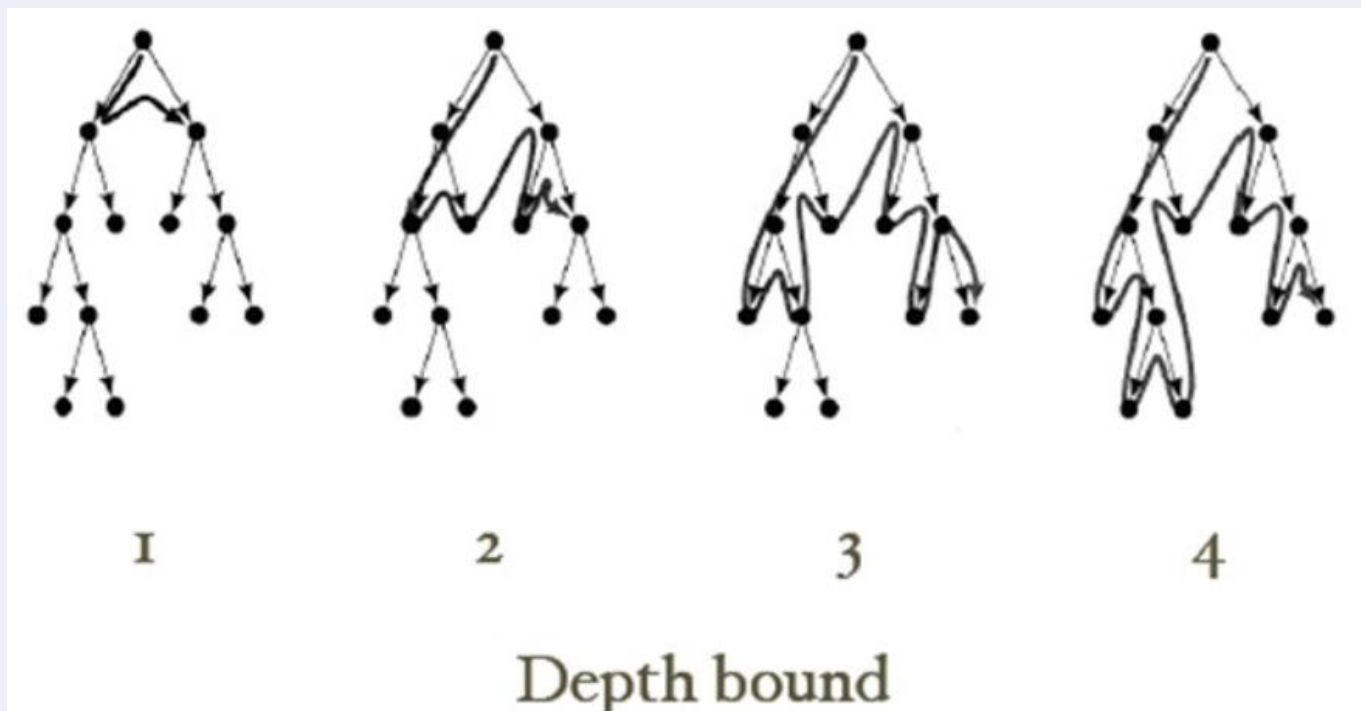


팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

Iterative Deepening Search



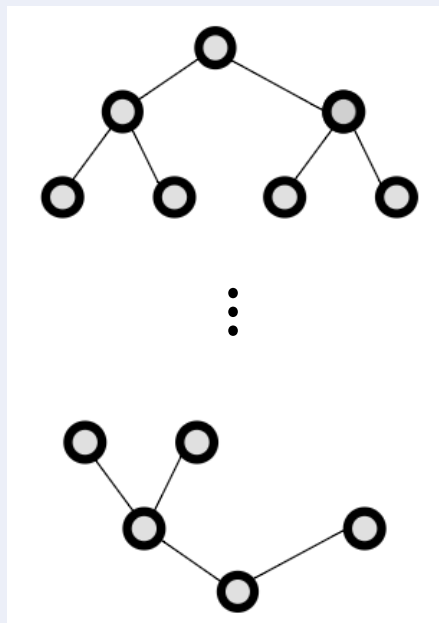


팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

Bidirectional search





팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Uninformed Search

알고리즘 간 비교

- ◇ b : branching factor
- ◇ d : depth of solution
- ◇ m : maximum depth of the search tree
- ◇ l : depth limit
- ◇ C^* : path cost of the optimal solution

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes ^a	Yes ^{a,b}	No	No	Yes ^a	Yes ^{a,d}
Time	$O(b^d)$	$O(b^{1+\lceil C^*/\epsilon \rceil})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lceil C^*/\epsilon \rceil})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes ^c	Yes	No	No	Yes ^c	Yes ^{c,d}



팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Informed Search

도메인에 대한 정보가 주어진 상태에서 목적지 노드를 찾는 것



$$f(n) = g(n) + h(n)$$

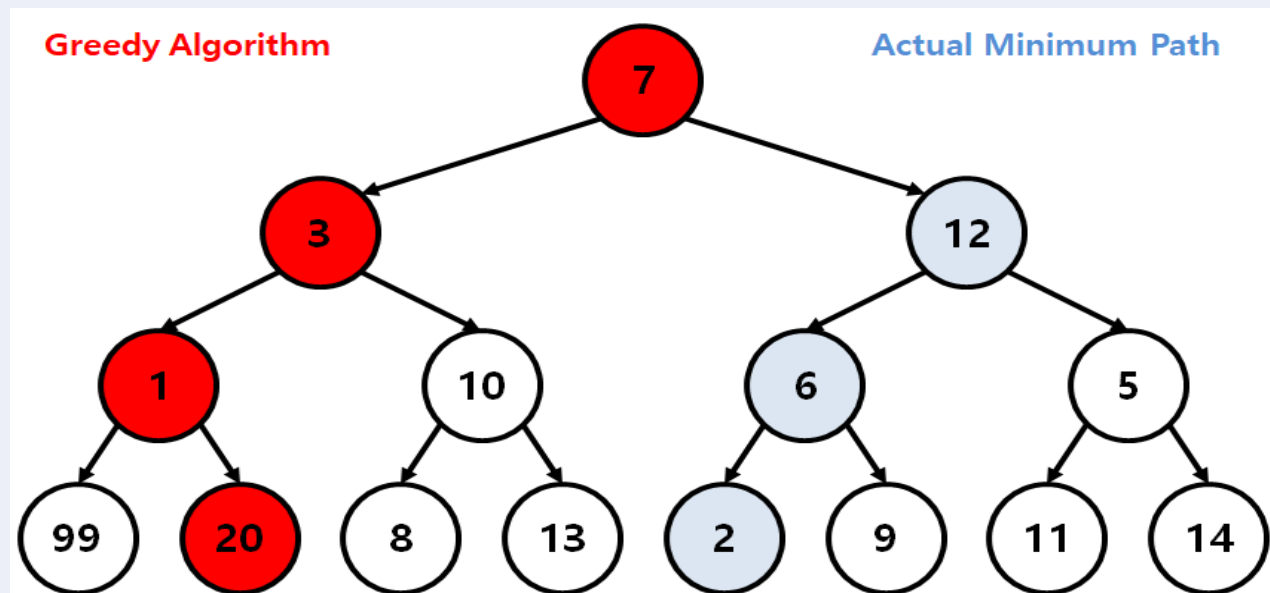


팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Informed Search

Best-first Search



$$f(n) = g(n) + h(n)$$

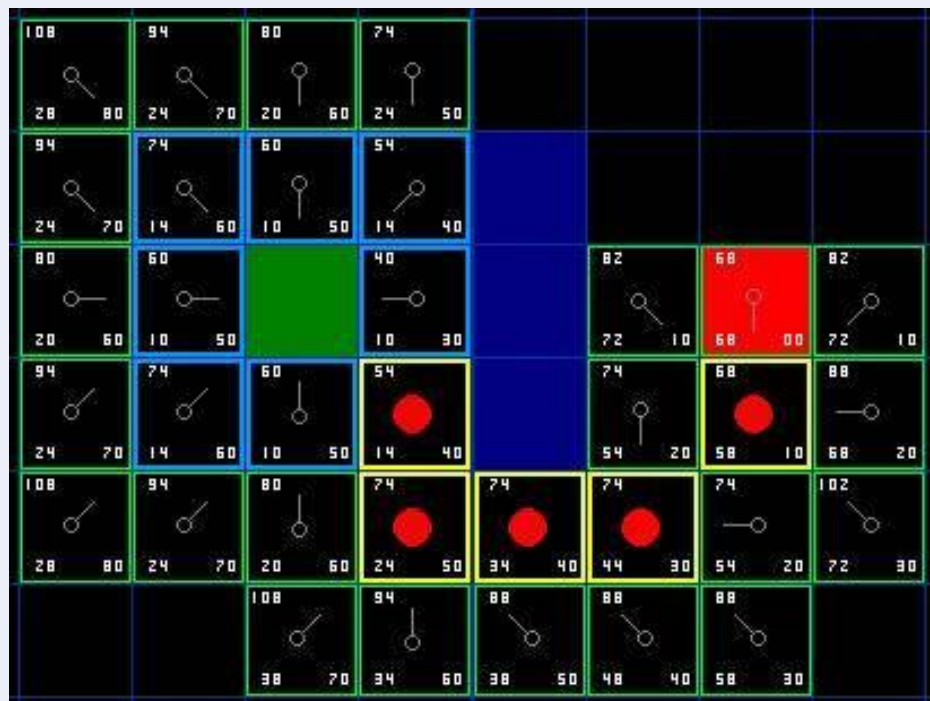


팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Informed Search

A* 알고리즘



$$f(n) = g(n) + h(n)$$



팀 프로젝트 '미로 찾기'

탐색 알고리즘 조사 및 분석

- Informed Search

그 밖에... 알파. A*의 한계와 Local search

A* 알고리즘의 한계는 가능한 모든 경우의 수를 전부 탐색해야 하는 특성에서 온다. → 미로가 커질 수록 느려짐.

→ 최적 해답을 포기하고 가능한 모든 경우를 전부 확인하지 않는다면 훨씬 빠른 탐색이 가능하다는 의미

Local search : 주변 노드만을 탐색하면서 해답을 찾아 나가는 방법

- Hill-climbing Search : 최적의 해를 찾아 이동
- Simulated Annealing Search : 상태가 개선되면 무조건 받아들임
- Genetic algorithm : 유전 법칙 이용



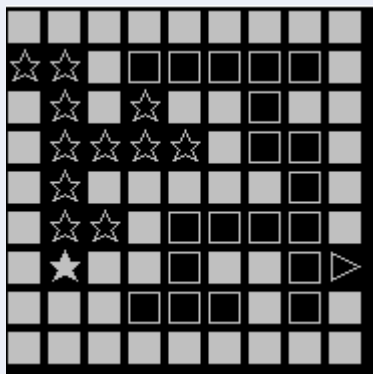
탐색 알고리즘 구현결과 확인



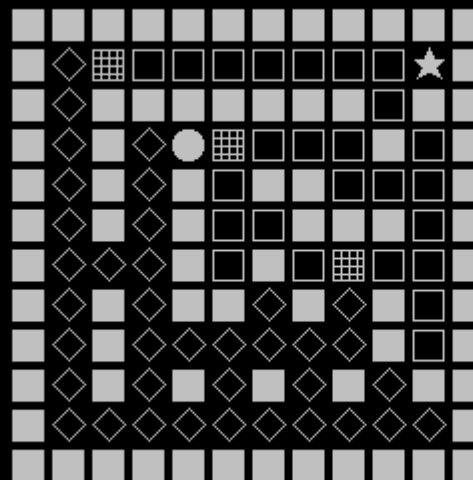
팀 프로젝트 '미로 찾기'

탐색 알고리즘 구현결과 확인

- BFS : 정주원



BFS 미로 탐색 알고리즘 동작



current_p.n : 11, q->length = 3
큐에 넣는 값 : [x,y,n] : [2, 0, 12]

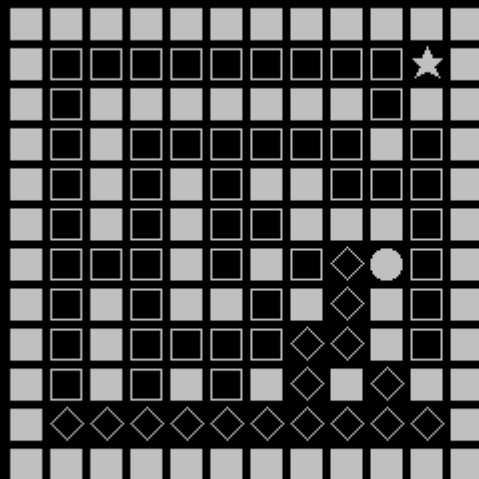


팀 프로젝트 '미로 찾기'

탐색 알고리즘 구현결과 확인

- DFS : 김제헌

DFS 미로 탐색 알고리즘 동작



current_p.n : 14, s->length = 5
스택에 넣는 값 : [x,y,n]: [7, 5, 12]

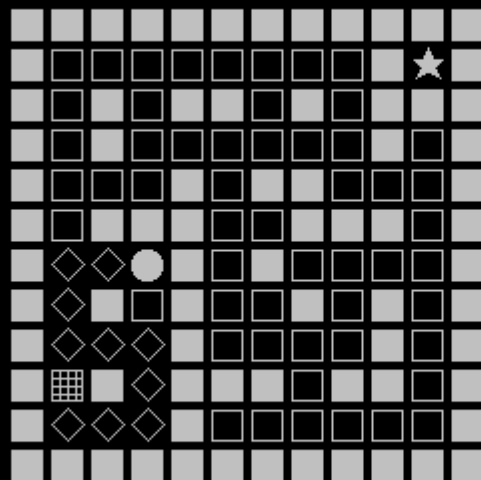


팀 프로젝트 '미로 찾기'

탐색 알고리즘 구현결과 확인

- DFS2 (실제미로 가정) : 김제헌

DFS2 [실제상황 가정] 미로 탐색 알고리즘 동작



current_p.n : 12, s->length = 4
스택에 넣는 값 : [x,y,n] : [0, 5, 9]

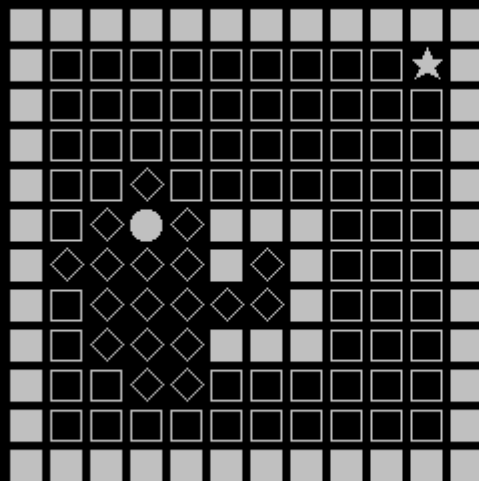


팀 프로젝트 '미로 찾기'

탐색 알고리즘 구현결과 확인

- Uniform-cost Search(Dijkstra) : 김제헌

Dijkstra 알고리즘 미로 탐색 알고리즘 동작



current_p.n : 31, h.heap_size = 10
힙트리에 넣는 값 : [x,y,n] : [0, 7, 32]

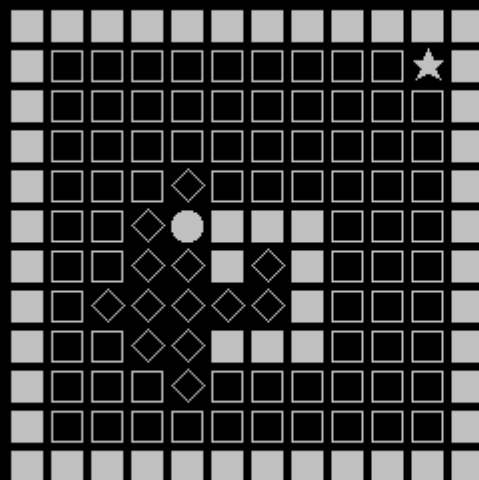


팀 프로젝트 '미로 찾기'

탐색 알고리즘 구현결과 확인

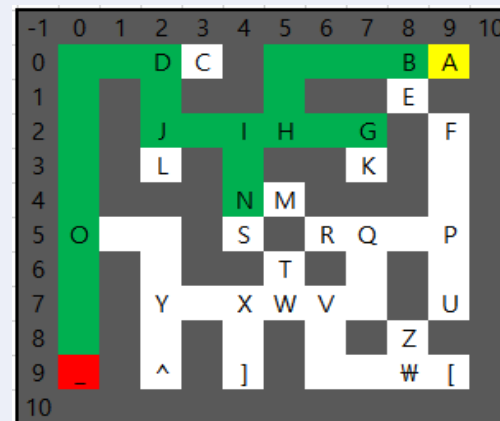
- A* Search : 김제헌

A * 알고리즘 미로 탐색 알고리즘 동작



current_p.n : 746, h.heap_size = 6
힉트리에 넣는 값 : [x,y,n]: [1, 5, 887]

- A* Search : 박상희





팀 프로젝트 '미로 찾기'

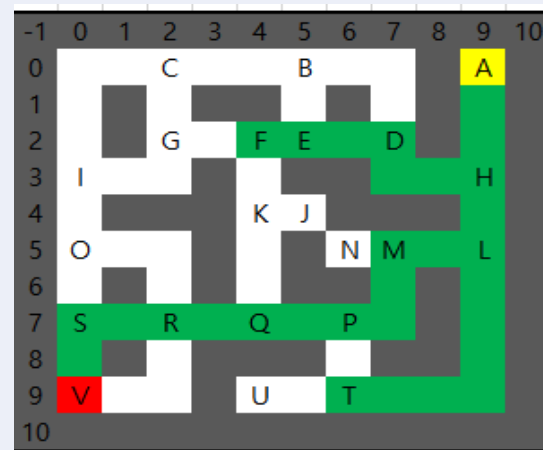
탐색 알고리즘 구현결과 확인

- A* Search : 박상희

```
graph의 인접리스트
A의 인접리스트 -> B
B의 인접리스트 -> H -> E -> A
C의 인접리스트 -> D
D의 인접리스트 -> O -> C -> J
E의 인접리스트 -> B
F의 인접리스트 -> P
G의 인접리스트 -> K -> H
H의 인접리스트 -> G -> I -> B
I의 인접리스트 -> N -> J -> H
J의 인접리스트 -> D -> L -> I
K의 인접리스트 -> G
L의 인접리스트 -> J
M의 인접리스트 -> N
N의 인접리스트 -> S -> M -> I
O의 인접리스트 -> Y -> -> D
P의 인접리스트 -> U -> F -> Q
Q의 인접리스트 -> P -> R -> V
R의 인접리스트 -> Q
S의 인접리스트 -> N
T의 인접리스트 -> W
U의 인접리스트 -> P
V의 인접리스트 -> Q -> W
W의 인접리스트 -> V -> T -> X
X의 인접리스트 -> W -> J -> Y
Y의 인접리스트 -> X -> -> O
Z의 인접리스트 -> W
[의 인접리스트 -> W
]의 인접리스트 -> -> Z -> V
'의 인접리스트 -> X
,의 인접리스트 -> Y
.의 인접리스트 -> O
```

```
현재 A노드 도착
Heap : [13.04, 1]
현재 B노드 도착
Heap : [14.60, 7] [102.00, 4]
현재 H노드 도착
Heap : [15.06, 8] [102.00, 4] [17.90, 6]
현재 I노드 도착
Heap : [15.40, 13] [16.28, 9] [17.90, 6] [102.00, 4]
현재 N노드 도착
Heap : [16.28, 9] [102.00, 4] [17.90, 6] [110.00, 18] [110.00, 12]
현재 J노드 도착
Heap : [17.90, 6] [20.22, 3] [110.00, 12] [110.00, 18] [102.00, 4] [110.00, 11]
현재 G노드 도착
Heap : [20.22, 3] [102.00, 4] [109.00, 10] [110.00, 18] [110.00, 11] [110.00, 12]
현재 D노드 도착
Heap : [27.85, 14] [110.00, 12] [102.00, 4] [110.00, 18] [110.00, 11] [109.00, 10] [112.00, 2]
현재 O노드 도착
Heap : [22.00, 30] [24.83, 24] [102.00, 4] [110.00, 12] [110.00, 11] [112.00, 2] [109.00, 10] [110.00, 18]
현재 _노드 도착
목적지 도착
```

- A* Search : 박상희





팀 프로젝트 '미로 찾기'

탐색 알고리즘 구현결과 확인

- A* Search : 박상희

graph2의 인접리스트

```
정점 A의 인접리스트 → H  
정점 B의 인접리스트 → E → C → D  
정점 C의 인접리스트 → I → G → B  
정점 D의 인접리스트 → E → B → H  
정점 E의 인접리스트 → F → B → D  
정점 F의 인접리스트 → K → G → E  
정점 G의 인접리스트 → I → C → F  
정점 H의 인접리스트 → L → D → A  
정점 I의 인접리스트 → O → G → C  
정점 J의 인접리스트 → K  
정점 K의 인접리스트 → Q → J → F  
정점 L의 인접리스트 → T → M → H  
정점 M의 인접리스트 → P → N → L  
정점 N의 인접리스트 → M  
정점 O의 인접리스트 → S → R → I  
정점 P의 인접리스트 → T → Q → M  
정점 Q의 인접리스트 → R → P → K  
정점 R의 인접리스트 → S → V → Q → O  
정점 S의 인접리스트 → V → R → O  
정점 T의 인접리스트 → U → P → L  
정점 U의 인접리스트 → T  
정점 V의 인접리스트 → S → R
```

```
현재 A노드 도착  
Heap : [13.82, 7]  
현재 H노드 도착  
Heap : [14.85, 11] [15.90, 3]  
현재 L노드 도착  
Heap : [15.06, 12] [18.00, 19] [15.90, 3]  
현재 M노드 도착  
Heap : [15.90, 3] [18.00, 19] [16.32, 15] [108.00, 13]  
현재 D노드 도착  
Heap : [16.32, 15] [16.60, 4] [108.00, 13] [18.00, 19] [20.30, 1]  
현재 P노드 도착  
Heap : [16.47, 16] [16.60, 4] [108.00, 13] [20.30, 1] [18.00, 19]  
현재 Q노드 도착  
Heap : [16.60, 4] [16.83, 17] [21.40, 10] [20.30, 1] [18.00, 19] [108.00, 13]  
현재 E노드 도착  
Heap : [16.83, 17] [18.00, 19] [17.06, 5] [20.30, 1] [108.00, 13] [21.40, 10]  
현재 R노드 도착  
Heap : [17.06, 5] [18.00, 19] [18.00, 18] [20.30, 1] [108.00, 13] [21.40, 10] [18.00, 21] [22.00, 14]  
현재 F노드 도착  
Heap : [18.00, 19] [18.28, 6] [18.00, 18] [20.30, 1] [108.00, 13] [21.40, 10] [18.00, 21] [22.00, 14]  
현재 T노드 도착  
Heap : [18.00, 18] [18.28, 6] [18.00, 21] [20.30, 1] [108.00, 13] [21.40, 10] [22.00, 14] [114.00, 20]  
현재 S노드 도착  
Heap : [18.00, 21] [18.28, 6] [21.40, 10] [20.30, 1] [108.00, 13] [114.00, 20] [22.00, 14]  
현재 V노드 도착
```



Q & A



Thank you!