

1.

균일비용 탐색(Uniform Cost Search)

- 경로 탐색에서 모든 간선의 가중치가 같을 때 사용하는 알고리즘입니다.
- 너비 우선 탐색과 유사하게 트리를 생성하며, 경로의 비용이 최소가 될 때까지 탐색합니다.
- 노드의 방문 순서에 따라 경로의 비용이 달라질 수 있습니다.

언덕오르기 탐색(Hill Climbing Search)

- 주어진 문제의 해결을 위해 현재 상태에서 가능한 한 가장 최적의 선택을 하는 방식으로 탐색합니다.
- 다른 경로를 모두 무시하고, 가장 기울기가 높은 방향으로만 이동합니다.
- 지역 최적해에 빠질 위험이 있으며, 해가 여러 개 존재할 경우 어느 하나를 찾을 수 없습니다.

A* 알고리즘(A* Search Algorithm)

- 경로 탐색에서 가중치가 있는 간선을 통해 최단 경로를 찾는 데 사용하는 알고리즘입니다.
- 경로의 비용과 현재 위치에서 목표 지점까지의 추정 비용을 고려하여 탐색합니다.
- 추정 비용은 휴리스틱 함수(Heuristic Function)를 사용하여 계산합니다.
- 최적의 해를 찾을 수 있으며, 많은 양의 메모리를 요구합니다.

각 알고리즘의 특성을 비교해보면, 균일비용 탐색은 간단하고 메모리 사용량이 적지만 경로 탐색에 시간이 오래 걸릴 수 있습니다.

언덕오르기 탐색은 경로 탐색 시간이 빠르지만 지역 최적해에 빠질 위험이 있습니다.

A* 알고리즘은 최적의 해를 찾을 수 있지만 메모리 사용량이 많이 필요하며, 휴리스틱 함수의 정확도에 따라 탐색 속도와 결과가 달라질 수 있습니다.

따라서, 경로 탐색에 적합한 알고리즘을 선택할 때는 문제의 성격과 요구 사항을 고려하여 적절한 알고리즘을 선택하는 것이 중요합니다.

2.

$$G(n) = 0$$

1	5	2
	4	3
7	8	6

$$F = 0 + 5 = 5$$

$$G(n) = 1$$

	5	2
1	4	3
7	8	6

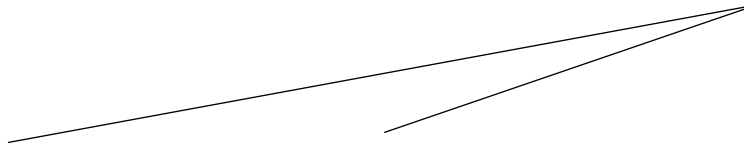
$$F = 1 + 6 = 7$$

1	5	2
7	4	3
	8	6

$$F = 1 + 6 = 7$$

1	5	2
4		3
7	8	6

$$F = 1 + 4 = 5$$



$$G(n) = 2$$

1		2
4	5	3
7	8	6

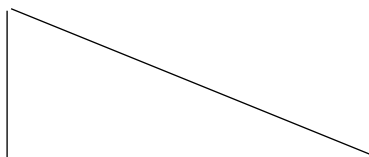
$$F = 2 + 3 = 5$$

1	5	2
4	3	
7	8	6

$$F = 2 + 4 = 6$$

1	5	2
4	8	3
7		6

$$F = 2 + 5 = 7$$



$$G(n) = 3$$

1	2	
4	5	3
7	8	6

$$F = 3 + 2 = 5$$

	1	2
4	5	3
7	8	6

$$F = 3 + 4 = 7$$



|

$G(n) = 4$

1	2	3
4	5	
7	8	6

$F = 4 + 1 = 5$

$G(n) = 5$

1	2	3
4	5	6
7	8	

$F = 5 + 0 = 5$