

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)

제 80강 - Player VS Computer 구성 및 알고리즘 이론 학습하기

Player VS Computer 구성 및 알고리즘 이론

보드 게임을 위한 인공지능 알고리즘

- 오목(Gomoku)은 한 쪽의 이익이 다른 한 쪽의 손실을 의미하는 제로섬(Zero-Sum) 게임입니다.
- 제로섬 게임에 활용되는 Minimax 알고리즘과 Alpha-Pruning 알고리즘의 동작 원리를 이해할 수 있습니다.

Player VS Computer 구성 및 알고리즘 이론

제로섬 알고리즘의 원리

- Minimax 알고리즘은 인공지능과 사람이 모두 최적에 가까운 수를 둔다는 가정을 두고 논리를 전개합니다.
- 어떠한 상황이 최적인 지를 구분해주는 역할을 수행하는 인공지능 내부의 평가 함수의 성능이 중요합니다.

Player VS Computer 구성 및 알고리즘 이론

평가 함수

- 평가 함수의 효율성이 높을 수록 강력한 인공지능이 탄생할 수 있습니다.
- 따라서 인공지능 개발자의 입장에서 오목 게임의 특징에 대해서 바르게 이해할 필요가 있습니다.
- 본 강좌에서는 오목에서 ‘오목’, ‘열린 사’, ‘열린 삼’, ‘닫힌 삼’ 등의 포석이 가지는 의미와 가치를 고려하여 오목 인공지능 알고리즘에 적용하는 방식의 평가 함수를 사용합니다.

Player VS Computer 구성 및 알고리즘 이론

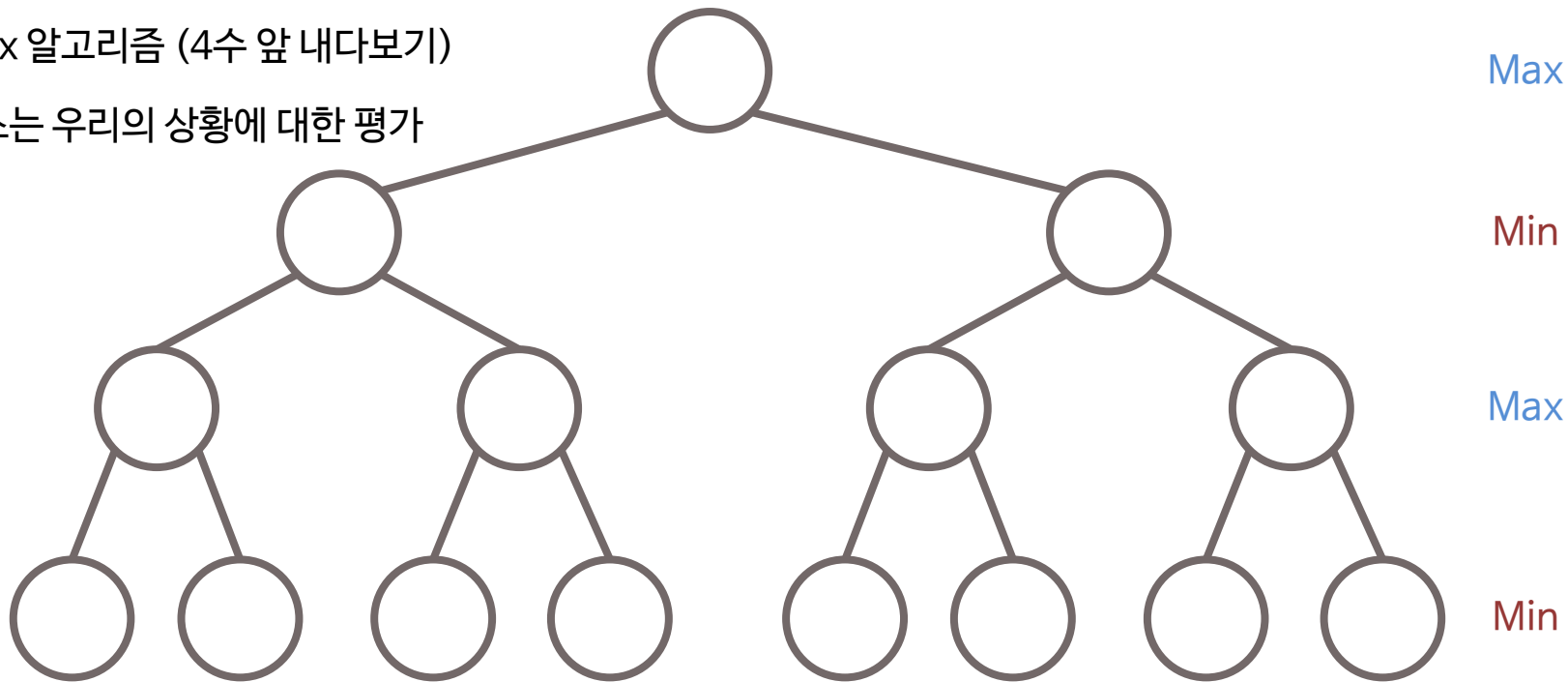
Minimax 알고리즘

- 2인 제로섬 게임(Zero-Sum Game)에서 가장 기본이 되는 알고리즘 중 하나입니다.
- Minimax 알고리즘은 정해진 Depth까지 노드의 가짓수를 탐색하는 알고리즘입니다.
- Minimax는 모든 플레이어가 합리적이라는 가정을 하므로, 우리는 우리의 상황에 대한 평가가 높은 경우만을 선택하고, 상대방은 우리의 상황에 대한 평가가 낮은 경우만을 선택한다고 가정합니다.
- 이러한 원리를 압축하여 우리를 Max, 상대방을 Min으로 정합니다.

Player VS Computer 구성 및 알고리즘 이론

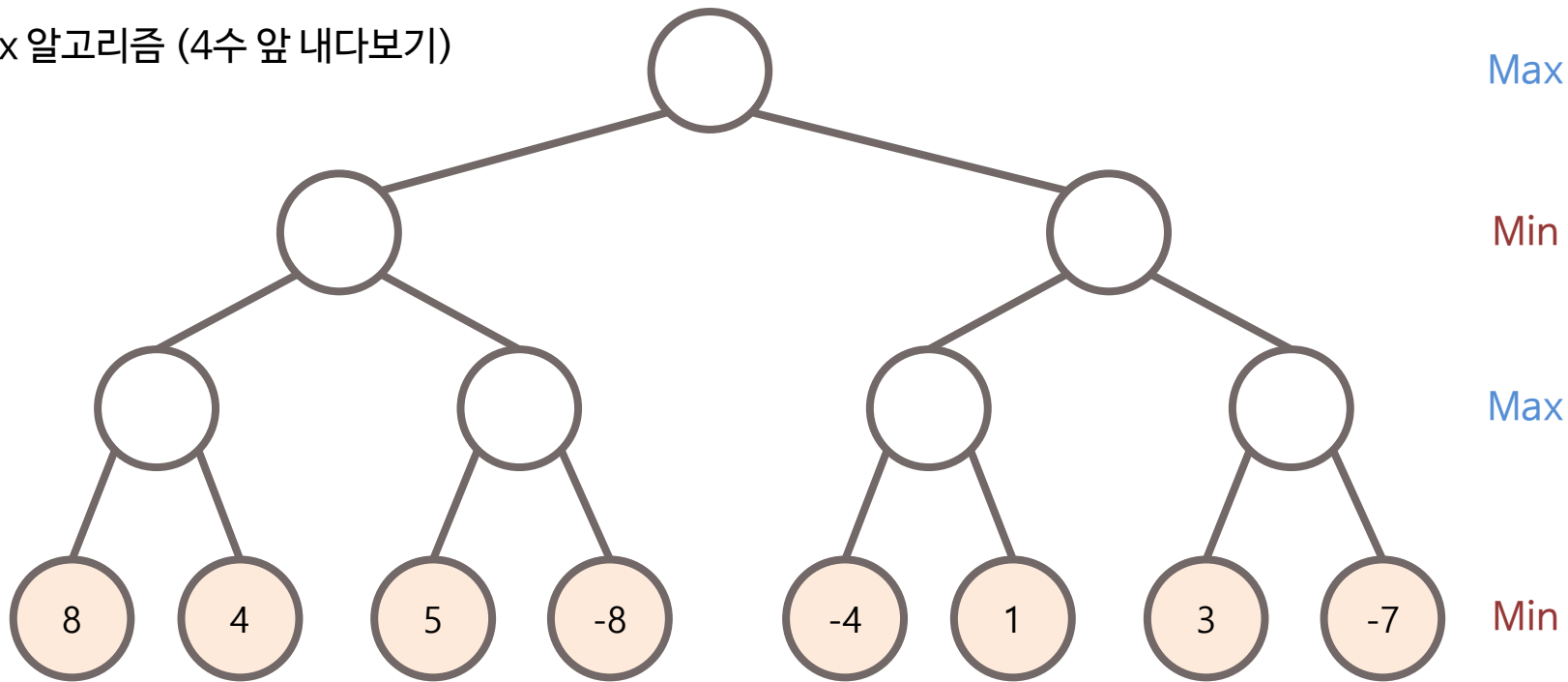
Minimax 알고리즘 (4수 앞 내다보기)

- 각 원소는 우리의 상황에 대한 평가



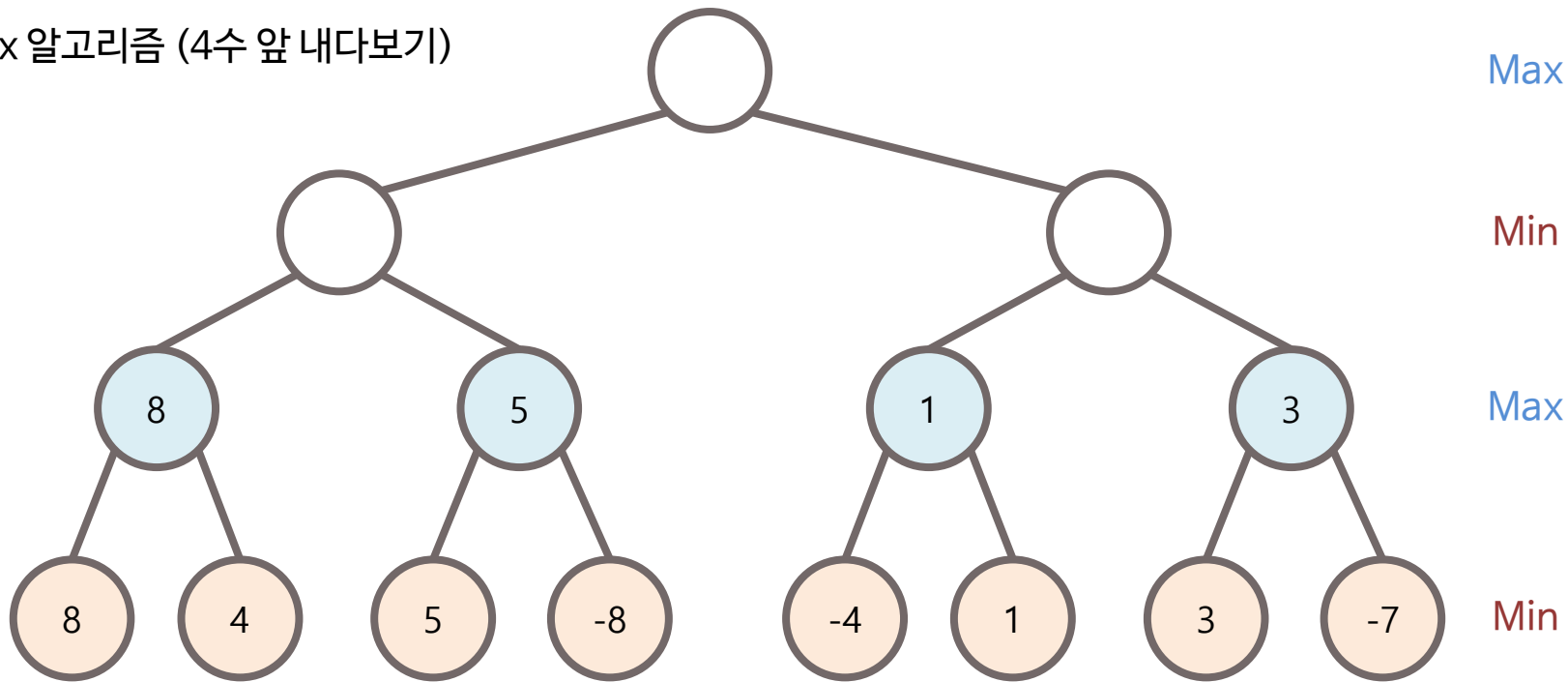
Player VS Computer 구성 및 알고리즘 이론

Minimax 알고리즘 (4수 앞 내다보기)



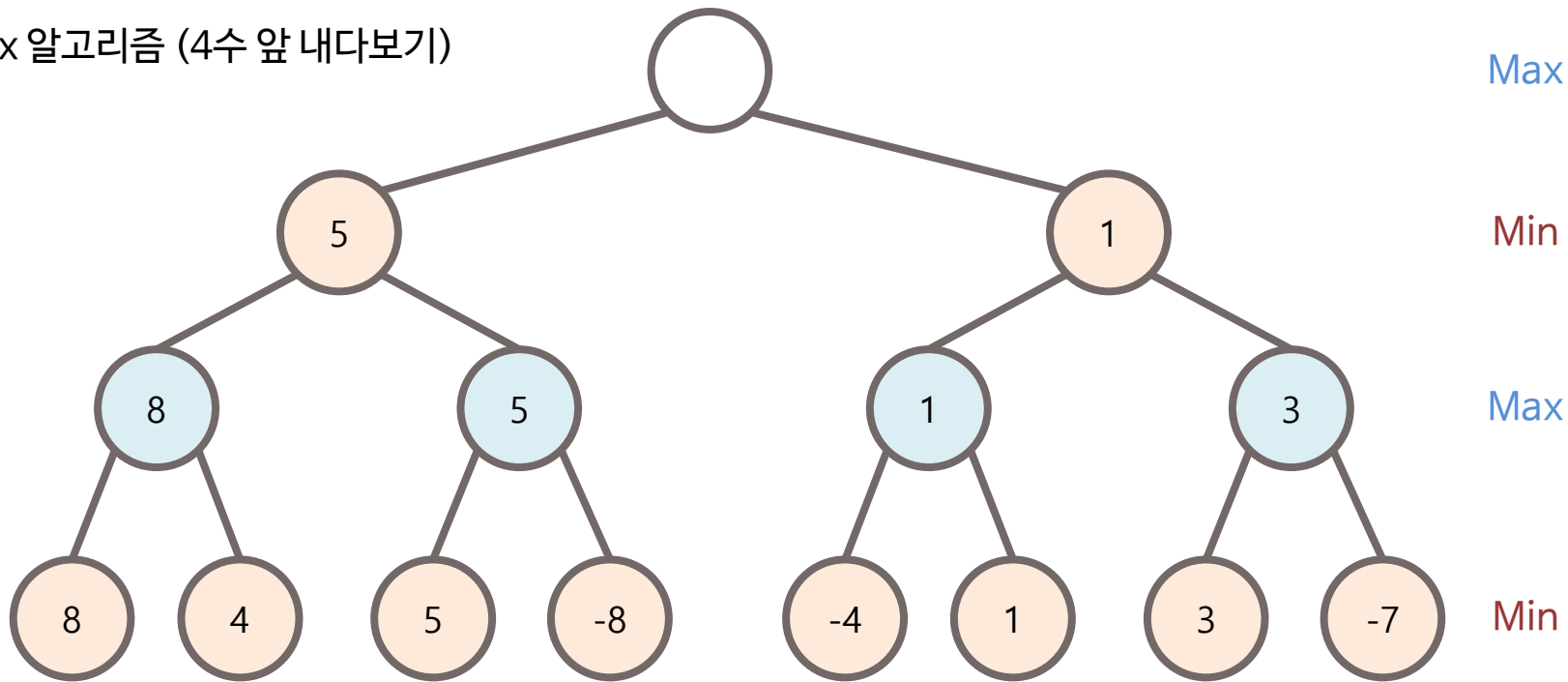
Player VS Computer 구성 및 알고리즘 이론

Minimax 알고리즘 (4수 앞 내다보기)



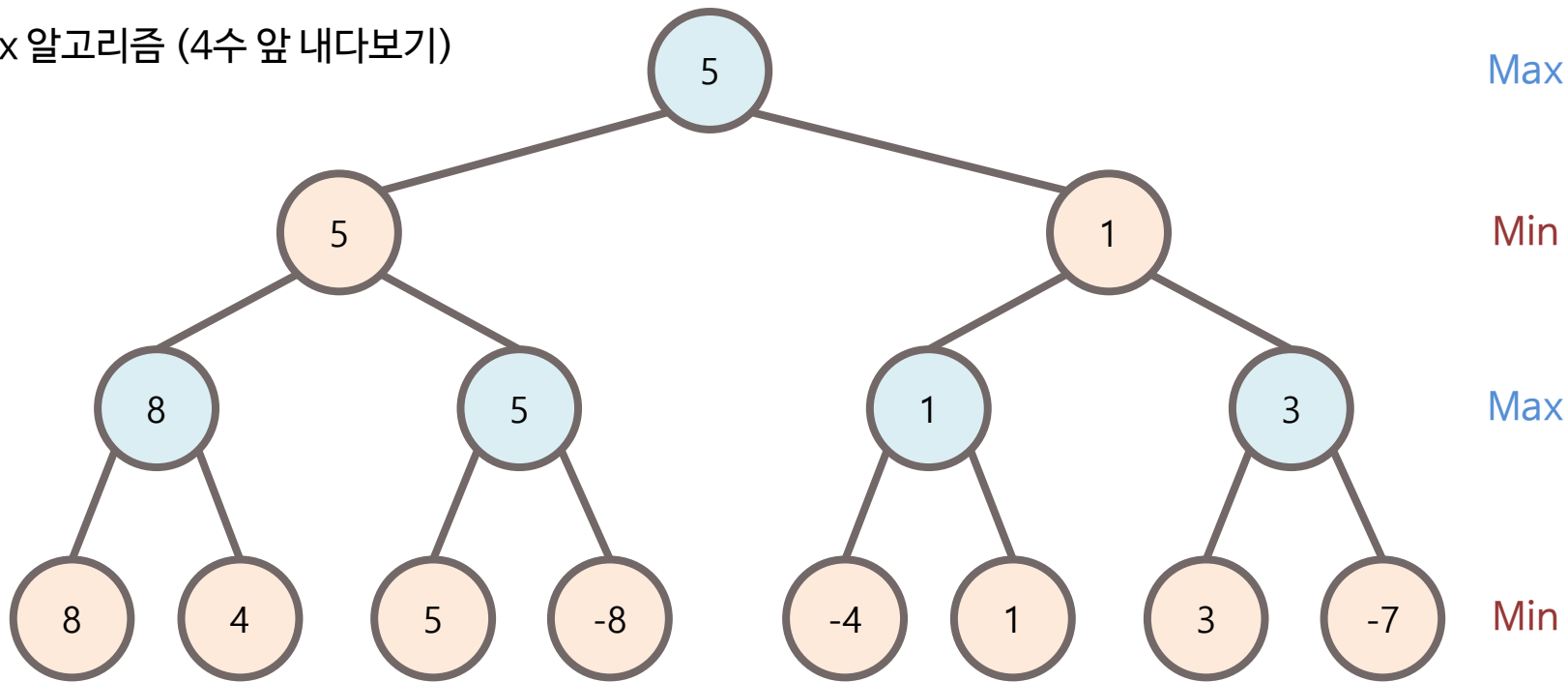
Player VS Computer 구성 및 알고리즘 이론

Minimax 알고리즘 (4수 앞 내다보기)



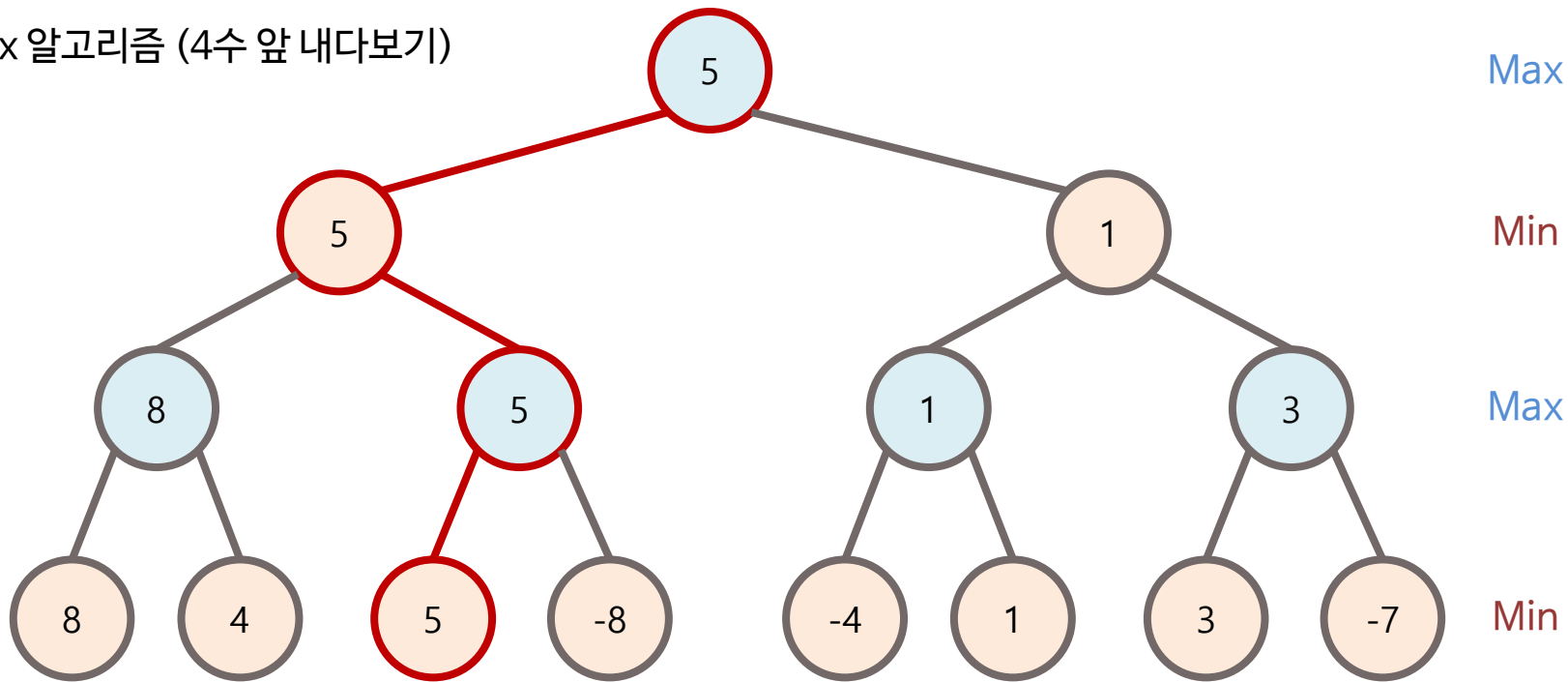
Player VS Computer 구성 및 알고리즘 이론

Minimax 알고리즘 (4수 앞 내다보기)



Player VS Computer 구성 및 알고리즘 이론

Minimax 알고리즘 (4수 앞 내다보기)



Player VS Computer 구성 및 알고리즘 이론

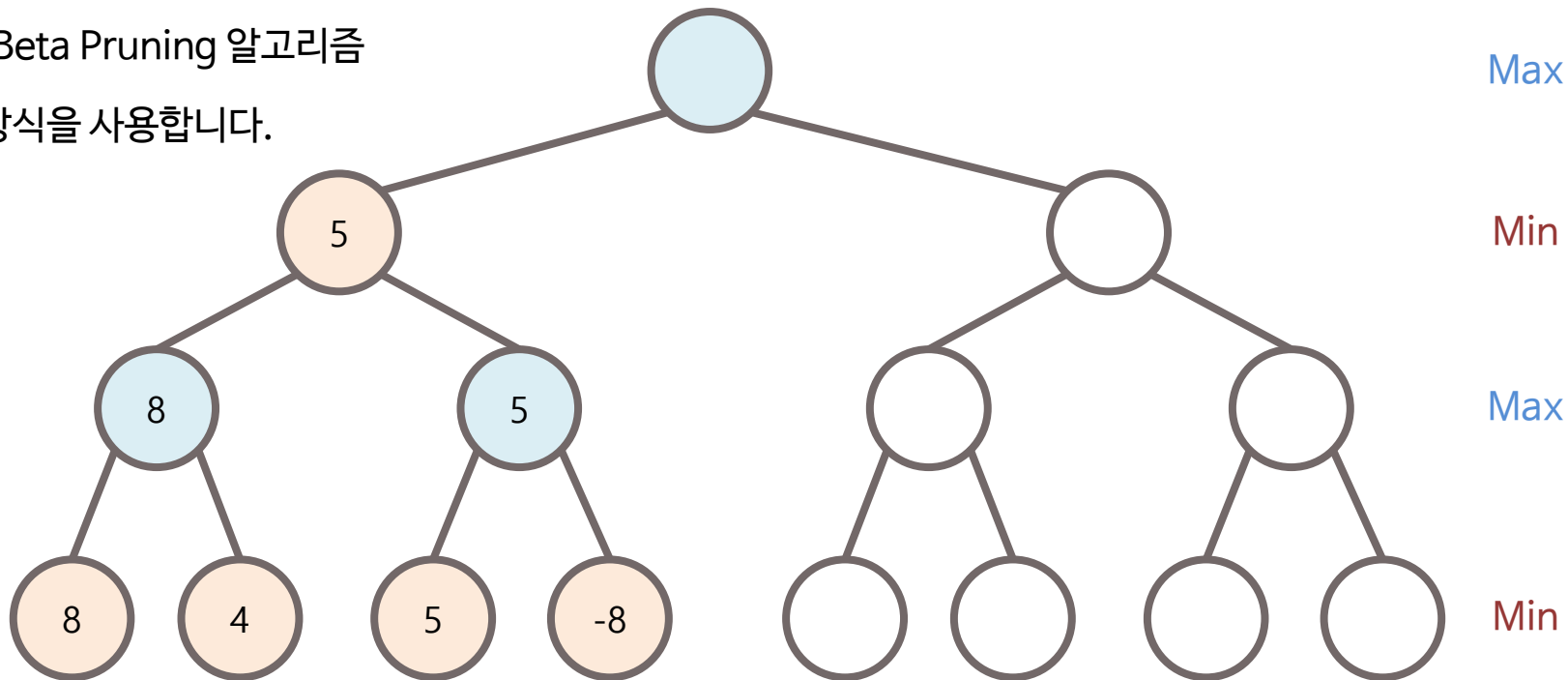
Alpha-Beta Pruning 알고리즘

- Minimax 알고리즘의 기반으로 효율성을 개선한 알고리즘입니다.
- 도달할 가능성이 없는 노드들은 탐색하지 않도록 가치를칩니다.
- EX) Max 노드의 Min 자식 노드 중 하나가 3이라는 값을 가진다면, Max 노드가 가지게 될 값은 항상 3 이상입니다. 이 때부터는 Max 노드의 Min 자식 노드 중에서 그 자식 Max 노드가 3 이하의 값을 가지는 노드가 하나라도 존재한다면 해당 Min 노드는 더 이상 탐색하지 않아도 됩니다.

Player VS Computer 구성 및 알고리즘 이론

Alpha-Beta Pruning 알고리즘

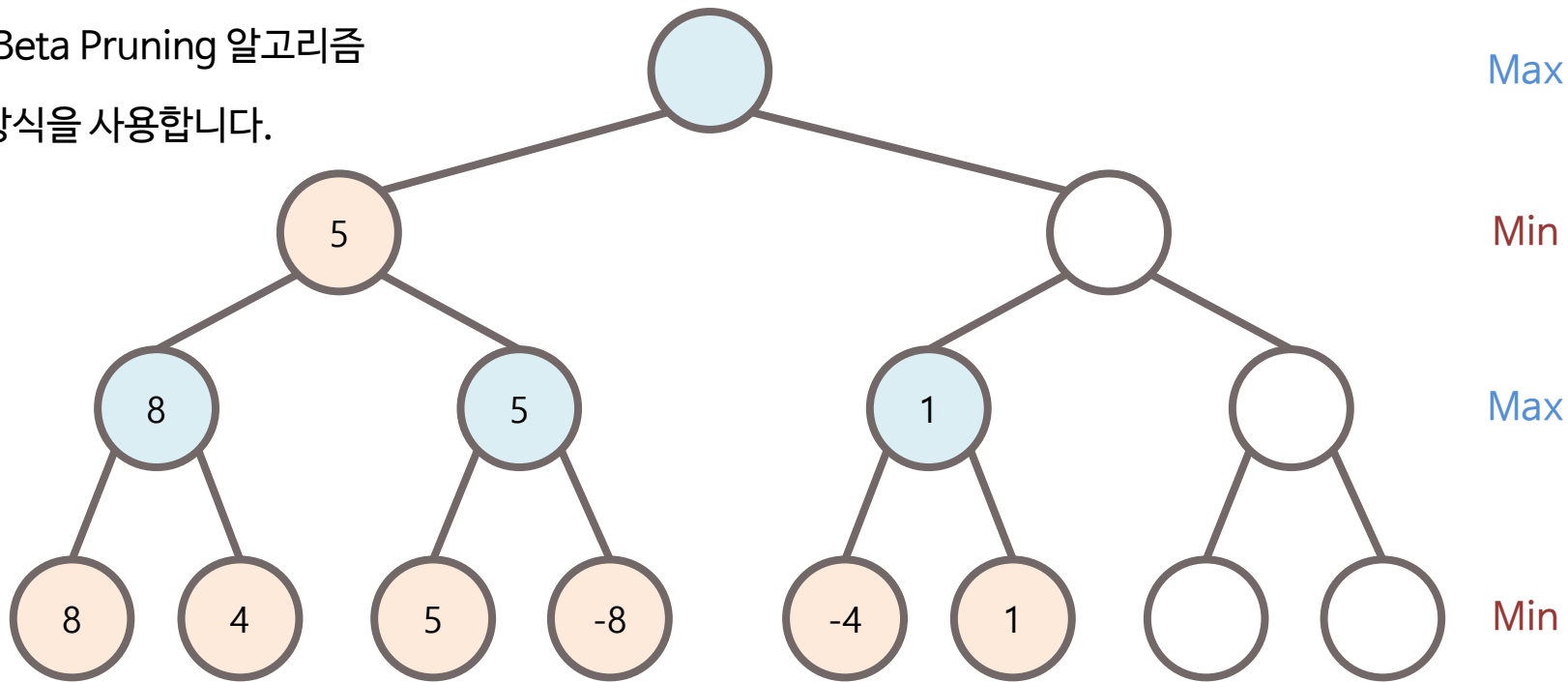
- DFS 방식을 사용합니다.



Player VS Computer 구성 및 알고리즘 이론

Alpha-Beta Pruning 알고리즘

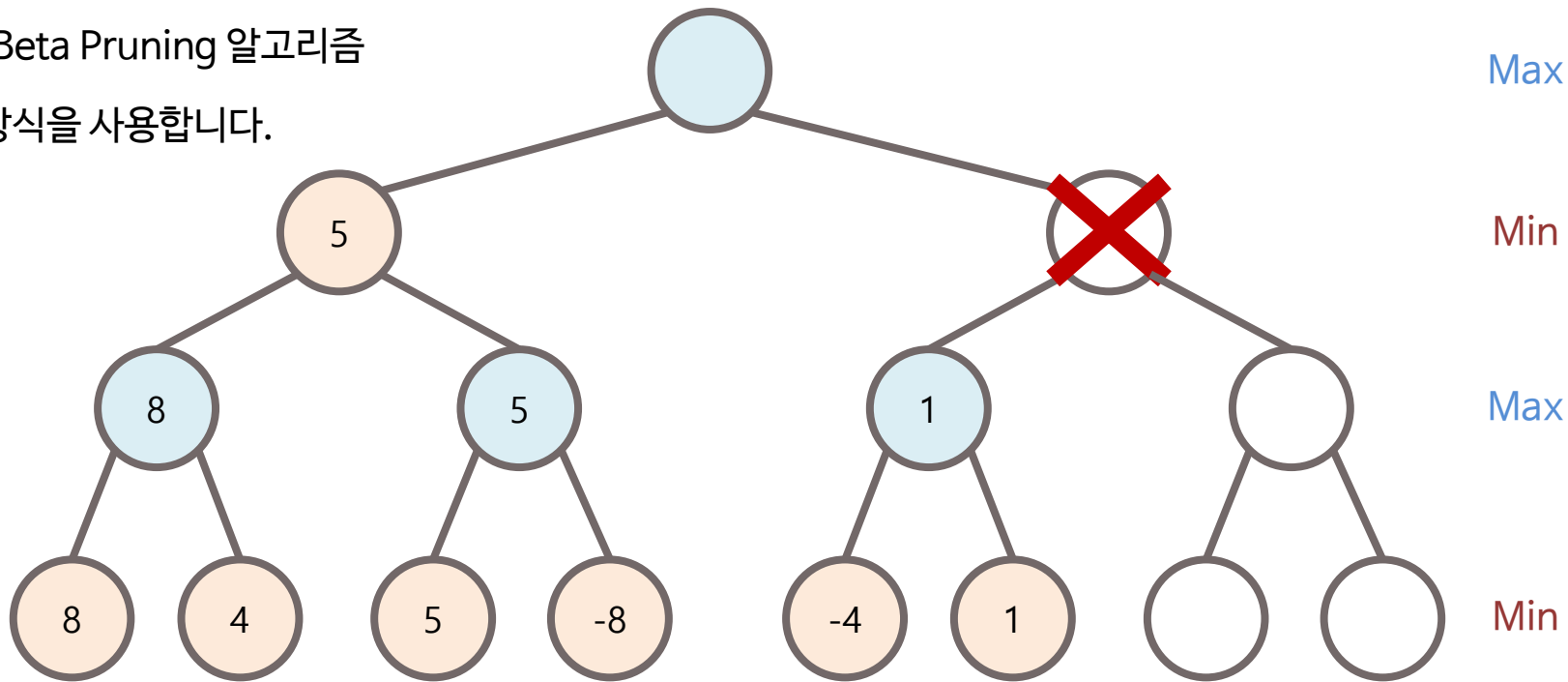
- DFS 방식을 사용합니다.



Player VS Computer 구성 및 알고리즘 이론

Alpha-Beta Pruning 알고리즘

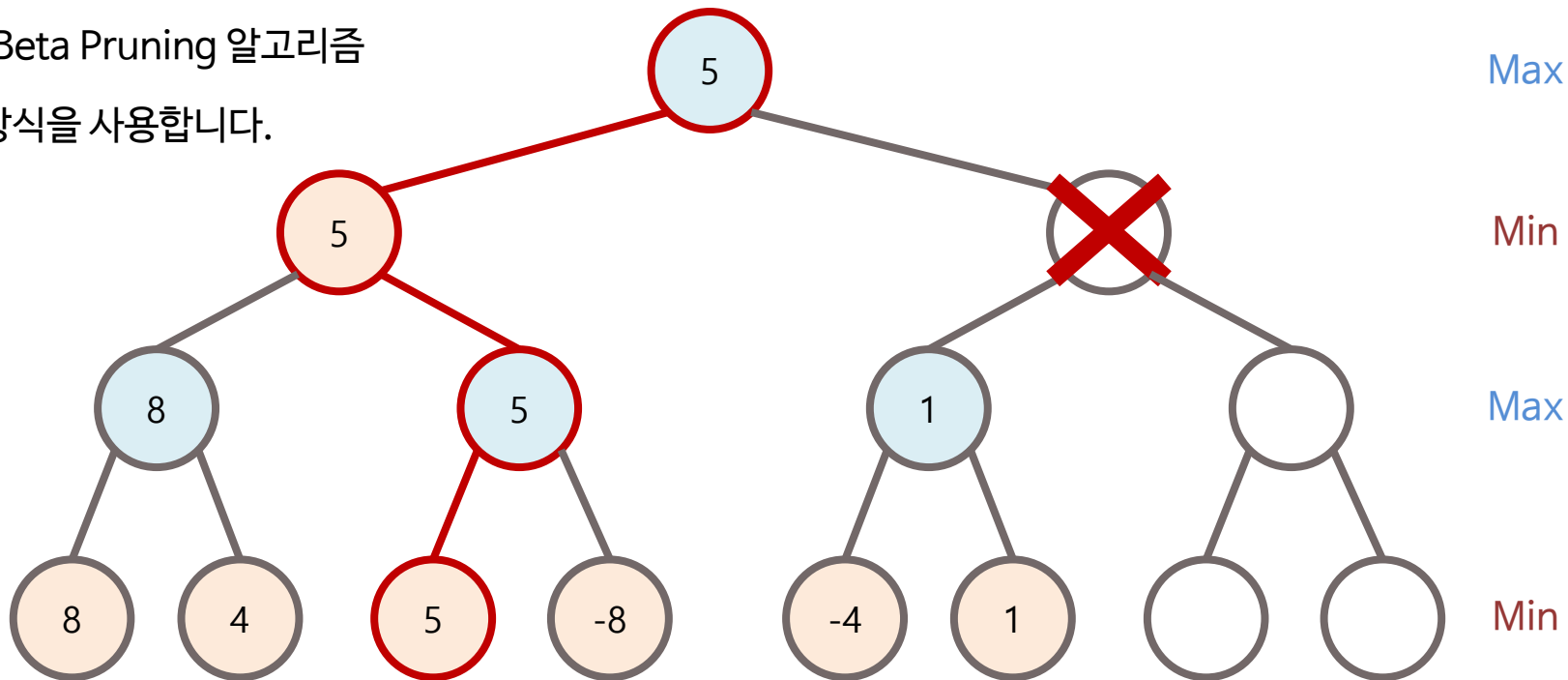
- DFS 방식을 사용합니다.



Player VS Computer 구성 및 알고리즘 이론

Alpha-Beta Pruning 알고리즘

- DFS 방식을 사용합니다.



Player VS Computer 구성 및 알고리즘 이론

평가 함수의 작성

- Minimax 알고리즘 및 Alpha-Beta Pruning 알고리즘은 정해진 작성 방법이 존재합니다.
- 하지만 평가 함수는 완벽한 해법이 정해져 있지 않으므로, 알고리즘 설계자가 임의로 설정할 수 있습니다.
- 평가 함수 알고리즘의 효율성 개선에 초점을 맞춘 연구가 필요합니다.

Player VS Computer 구성 및 알고리즘 이론

오목 게임에서의 평가 함수

- 열린 사가 닫힌 삼보다 강력한 수라는 것은 실험적으로 증명할 필요 없이 당연합니다.
- 하지만 열린 삼의 가치가 닫힌 사보다 가치로운지 등의 내용에 대해서는 깊게 고민할 필요가 있습니다.
- 특정 상황에 대한 가치 평가를 정량화 할 필요가 있습니다.

Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 1) 오목: 1,000,000



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 2) 열린 사: 200,000



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 3) 하나 닫힌 사: 15,000



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 4) 열린 삼: 4,000



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 5) 닫힌 사: 1,500



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 6) 하나 닫힌 삼: 1,500



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 7) 열린 이: 1,500



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 8) 닫힌 삼: 300



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 9) 하나 달한 이: 300



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

우선 순위 10) 닫힌 이: 50



Player VS Computer 구성 및 알고리즘 이론

특정한 돌의 패턴에 대한 가치 산정

추가 평가 요소: 바둑판의 천원점과 가까울 수록 점수 가산하기

Player VS Computer 구성 및 알고리즘 이론

실제 구현의 어려움

- 실제로 깊이 우선 탐색(DFS) 방식으로 게임 내에서 Alpha-Beta Pruning 기법을 구현하는 것은 어렵습니다.
- 따라서 자신의 상황에 맞게 몇 일 이상 연구하면서 알고리즘을 적용하는 방법이 추천됩니다.