

# 제4강 탐색하는 기계

<인공지능 입문> 강의 노트

장 병 탁

서울대학교 컴퓨터공학부 &  
인지과학/뇌과학 협동과정

<http://bi.snu.ac.kr/~btzhang/>

Version: 20180326

# 목차

문제해결과 탐색 .....	3
무정보 탐색 .....	6
경험적 탐색 .....	11
A* 알고리즘 .....	14
게임 트리 탐색 .....	15
알파베타 알고리즘 .....	17
Reading Assignments .....	22

# 8-퍼즐과 문제 해결

2	8	3
1	6	4
7		5

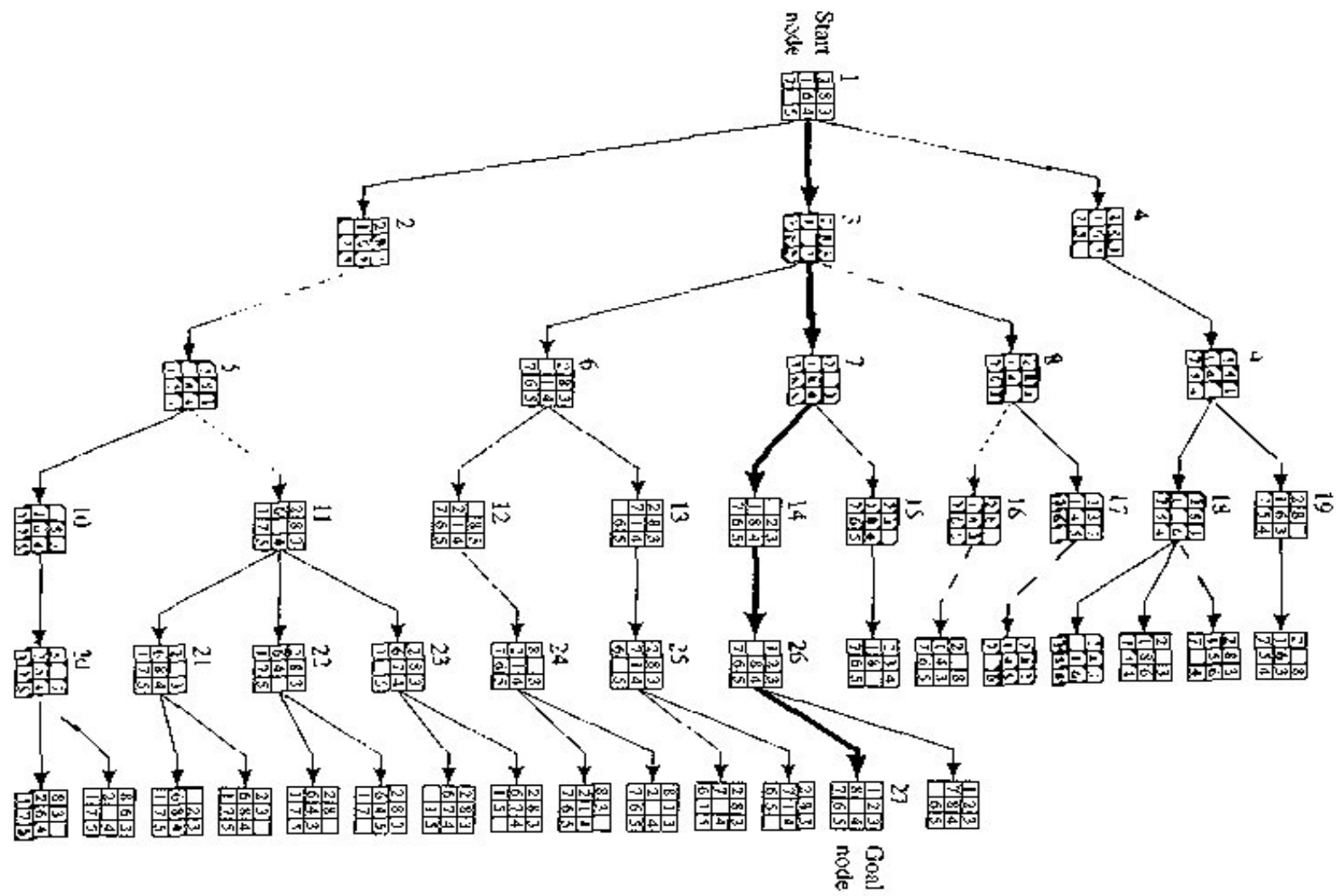


1	2	3
8		4
7	6	5

초기 상태

목표 상태

# 탐색 트리



# 문제해결과 탐색

- 문제 해결(problem solving)을 상태 공간 탐색(state space search)의 문제로 봄
- 상태 공간(state space): 가능한 모든 문제의 집합, 예: 모든 가능한 퍼즐판 모양의 집합
  - 상태  $s = (x_1, x_2, \dots, x_8, x_9)$ ,  $x_i = \{1, 2, \dots, 8, \text{blank}\}$
  - 연산자  $a$
  - 상태 전이(state transition):  $s_{n+1} = T(s_n, a)$
  - 상태공간 그래프
- 초기 상태(initial state): 주어진 문제
- 목표 상태(goal state): 문제가 해결된 상태
- 문제 해결 과정: 초기 상태에서 출발하여 목표 상태에 도달하는 경로. 또는 연산자들의 순서.
- 해공간(solution space): (목표 상태가 여러개일 경우) 모든 목표 상태의 집합.

# 탐색 알고리즘

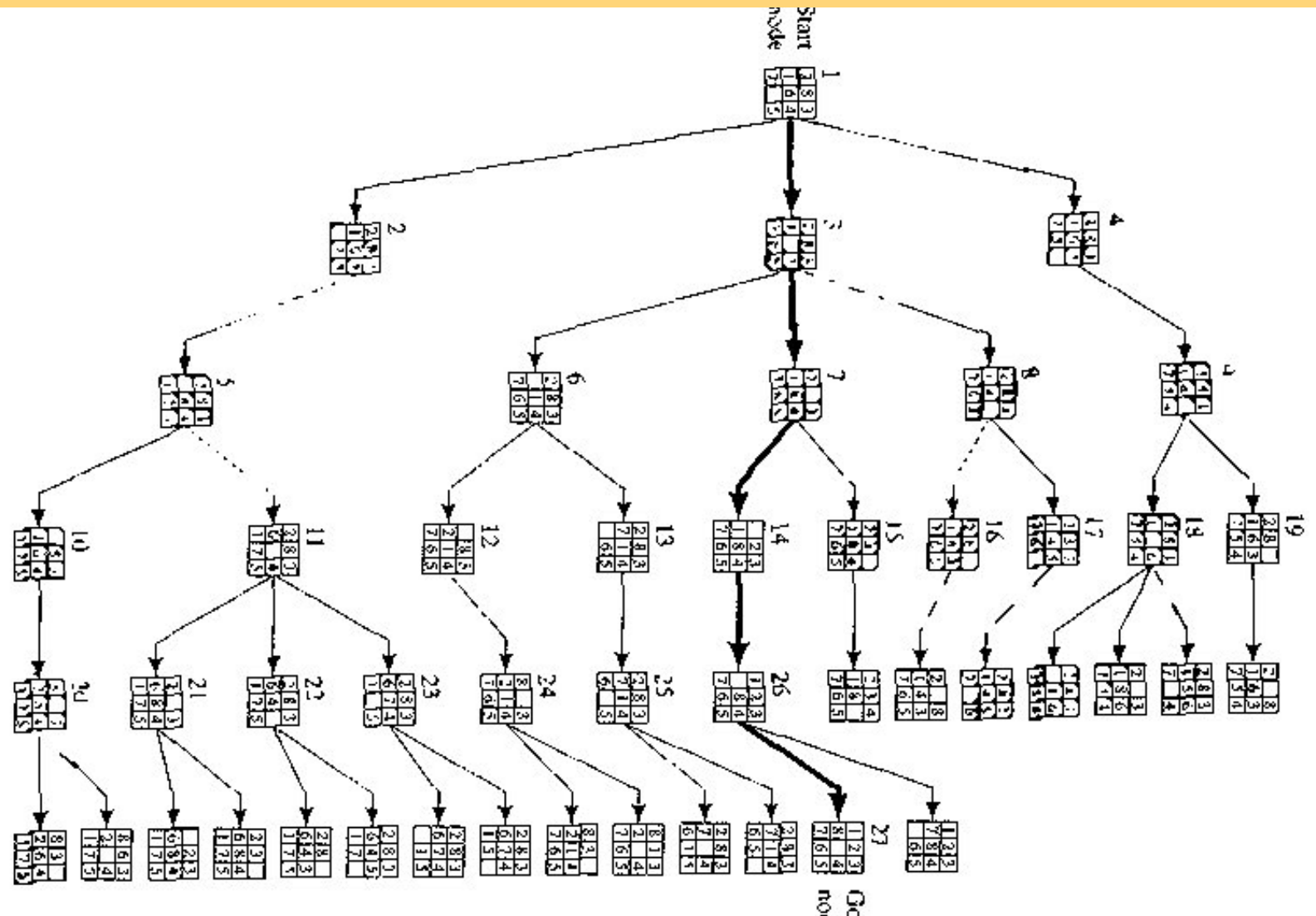
## □ 무정보 탐색

- 너비우선 탐색(Breadth-First Search, BFS)
- 깊이우선 탐색(Depth-First Search, DFS)
- 반복적 깊이우선 탐색(Iterative Deepening Search, IDS)

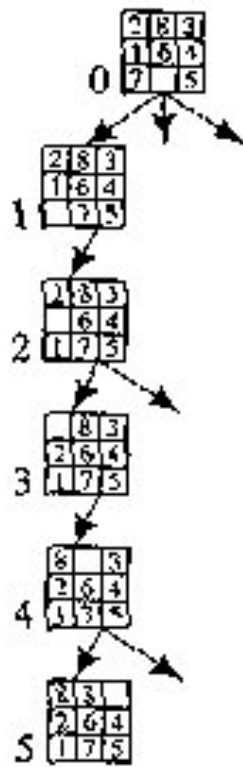
## □ 정보기반 탐색(경험적 탐색)

- 최선우선 탐색(Best-First Search)
- A\* 알고리즘
- 알파베타 알고리즘(게임)

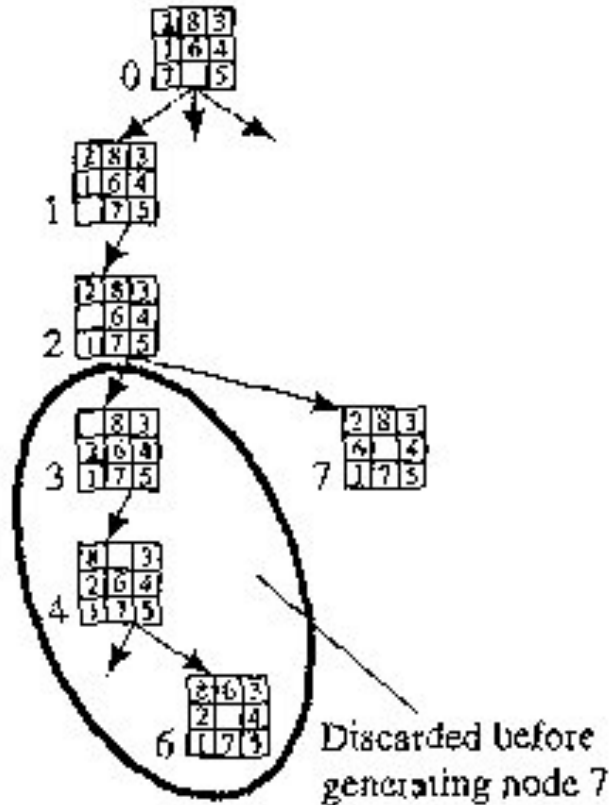
# 무정보 탐색: 너비우선 탐색



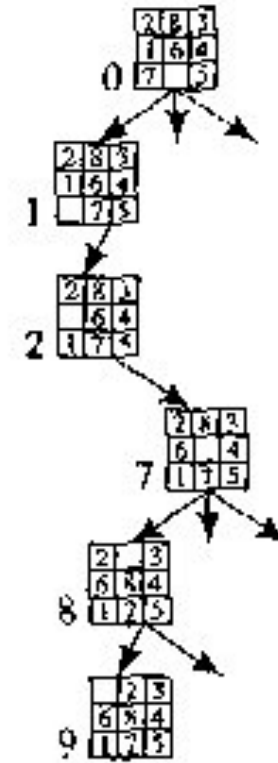
# 무정보 탐색: 깊이우선 탐색



(a)



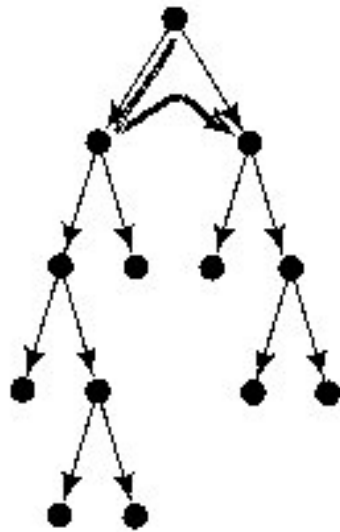
(b)



(c)



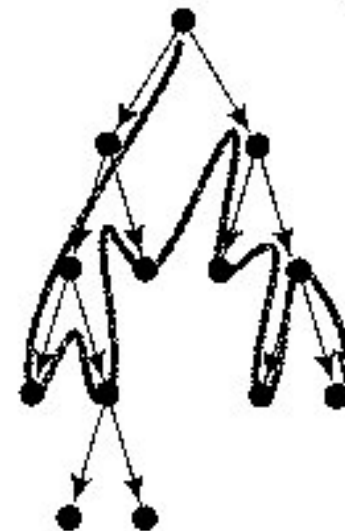
# 무정보 탐색: 반복적 깊이우선 탐색



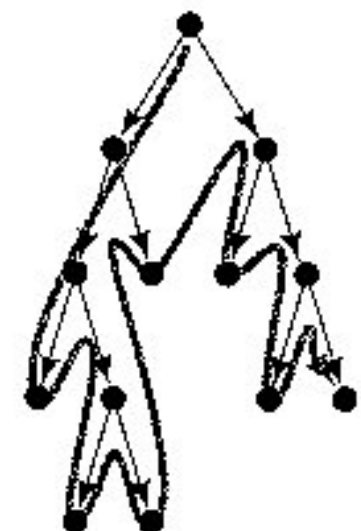
Depth bound = 1



Depth bound = 2



Depth bound = 3



Depth bound = 4

# 탐색 시간 복잡도 비교

- 방문한 노드 수
  - 너비우선 탐색

$$N_{bf} = 1 + b + b^2 + \dots + b^d = \frac{b^{d+1} - 1}{b - 1} \quad (b : \text{branching factor}, d : \text{depth})$$

- 반복적 깊이우선 탐색

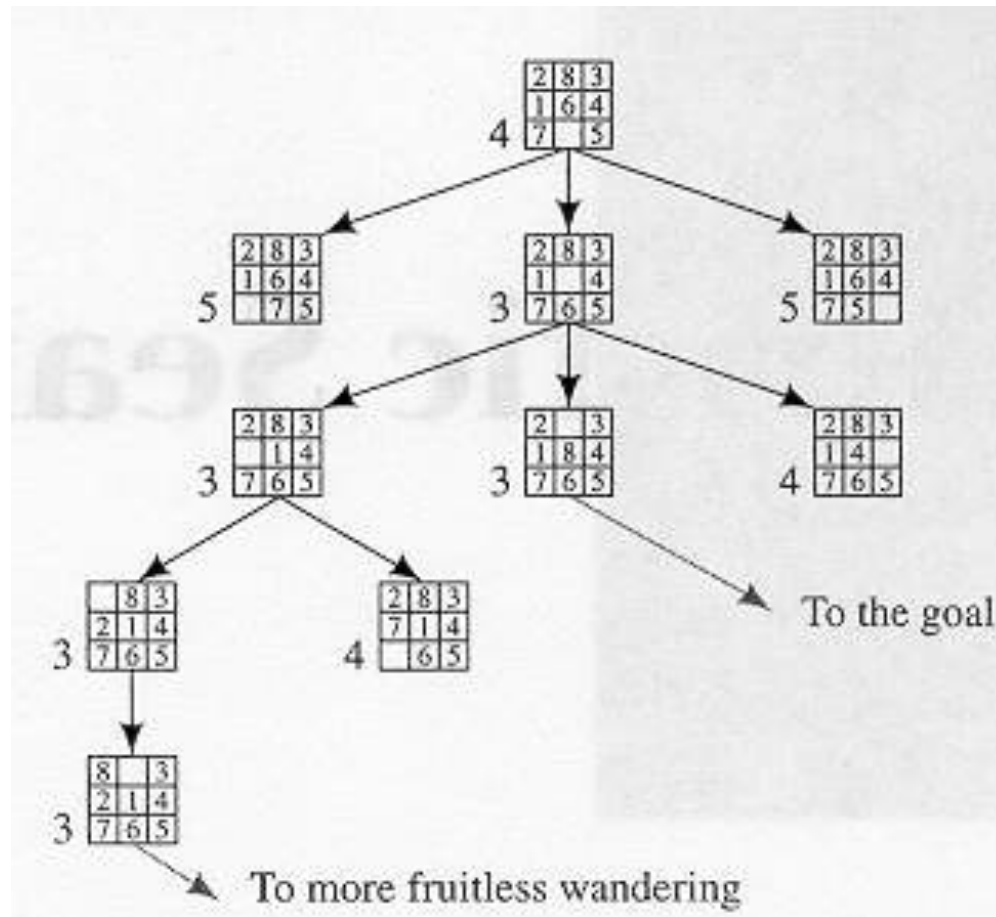
$$N_{df_j} = \frac{b^{j+1} - 1}{b - 1} : \text{number of nodes expanded down to level } j$$

$$\begin{aligned} N_{id} &= \sum_{j=0}^d \frac{b^{j+1} - 1}{b - 1} \\ &= \frac{1}{b - 1} \left[ b \left( \sum_{j=0}^d b^j \right) - \sum_{j=0}^d 1 \right] = \frac{1}{b - 1} \left[ b \left( \frac{b^{d+1} - 1}{b - 1} \right) - (d + 1) \right] \\ &= \frac{b^{d+2} - 2b - bd + d + 1}{(b - 1)^2} \end{aligned}$$

# 경험적 탐색

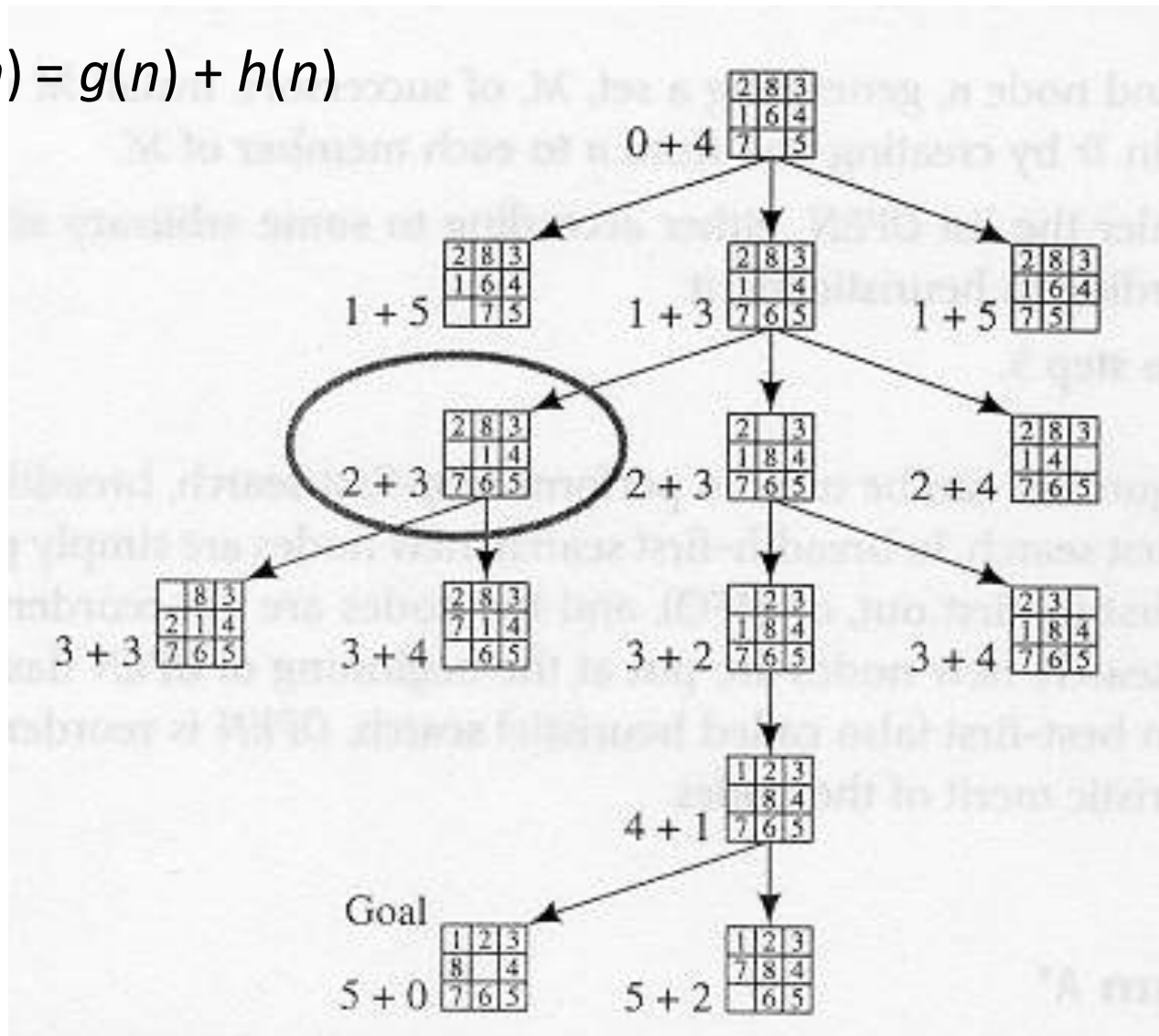
# (경험적) 평가 함수

(8퍼즐) 평가함수의 예: 목표 상태와 일치하지 않는 격자의 수



# 8-퍼즐과 문제 해결

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



## A\* 알고리즘

경험적 탐색을 하여도 평가함수가 허용성(admissibility)을 만족하면 최적의 해를 찾을수 있음을 보장함!

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

$$h(n) \leq h^*(n)$$

단조증가성(monotonicity) 증명:

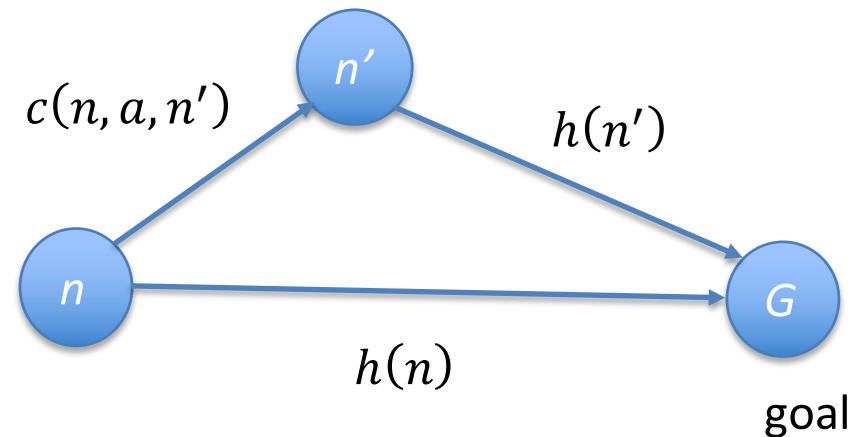
$$h(n) \leq c(n, a, n') + h(n')$$

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

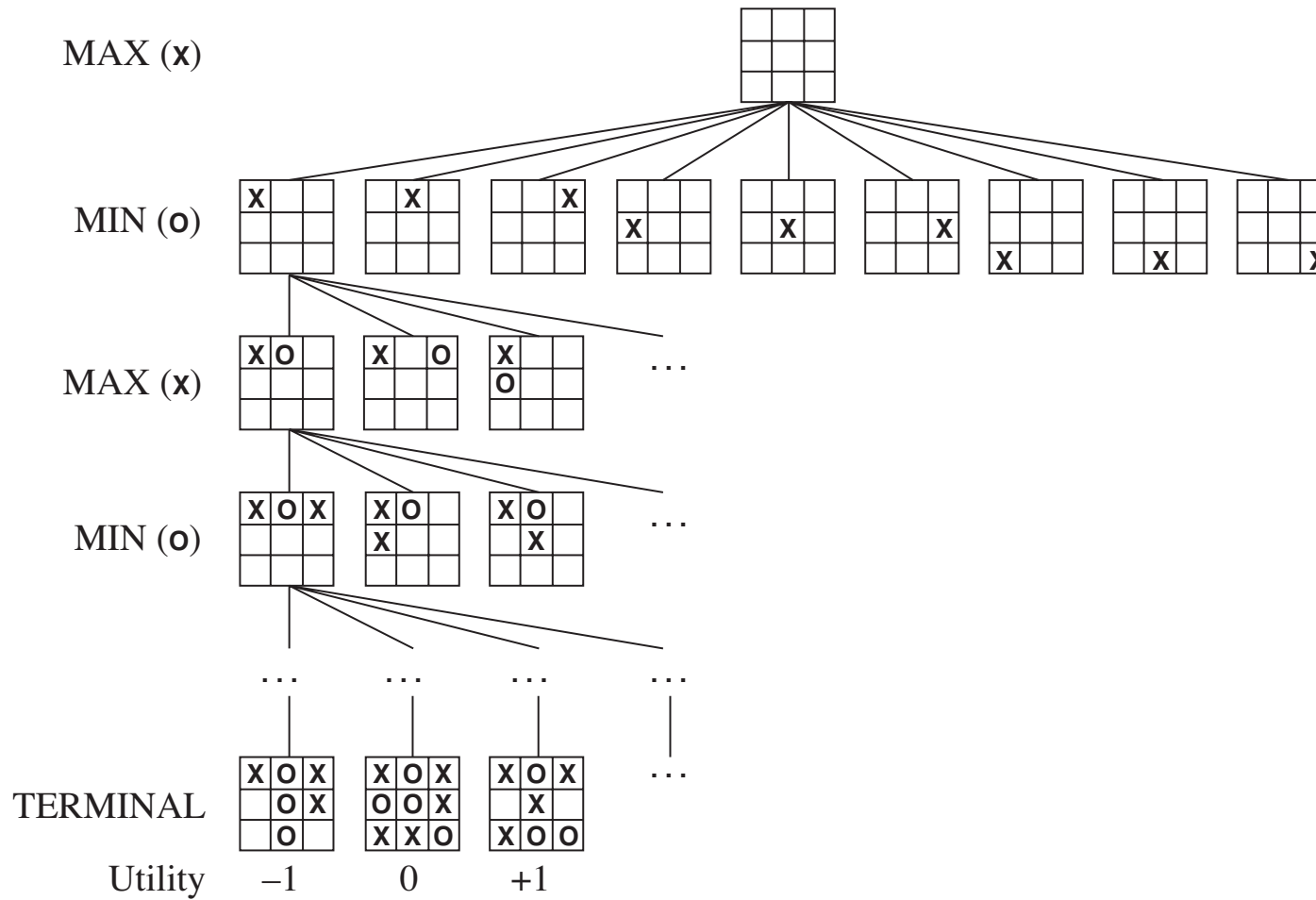
$$= g(n) + c(n, a, n') + h(n')$$

$$\geq g(n) + h(n)$$

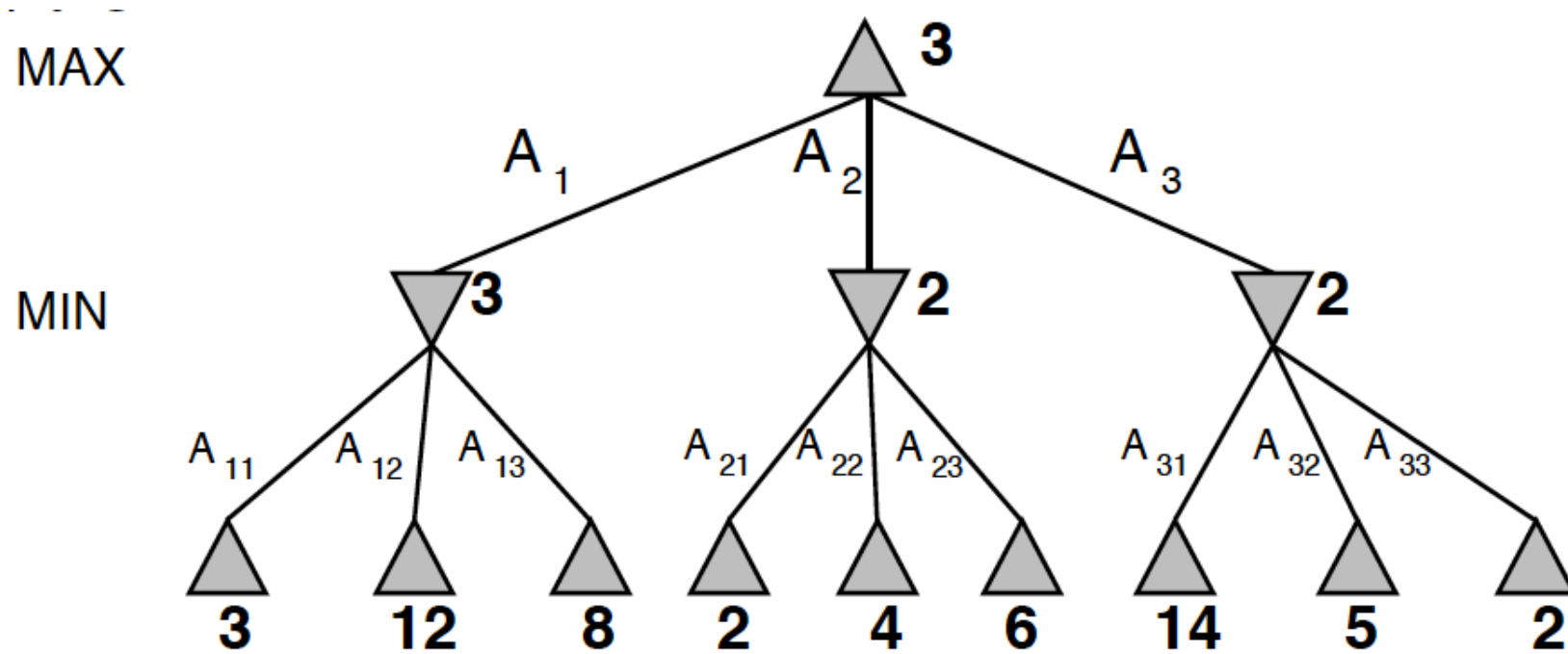
$$= f(n)$$



# 적대적 탐색: 게임 트리

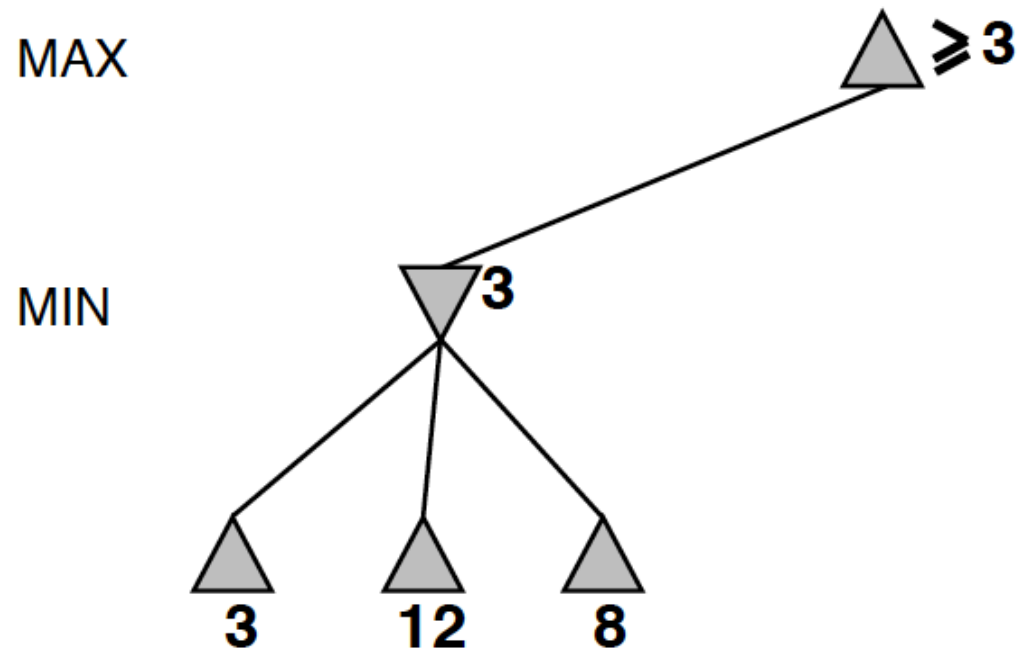


# 적대적 탐색: Minimax 알고리즘

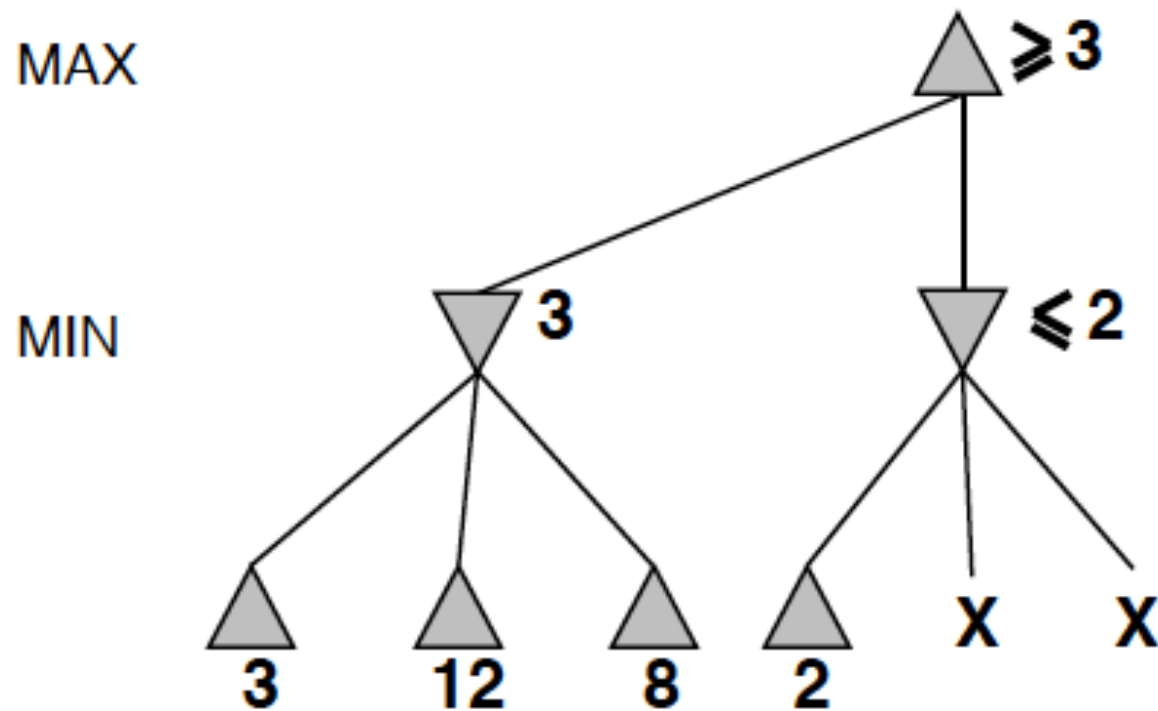




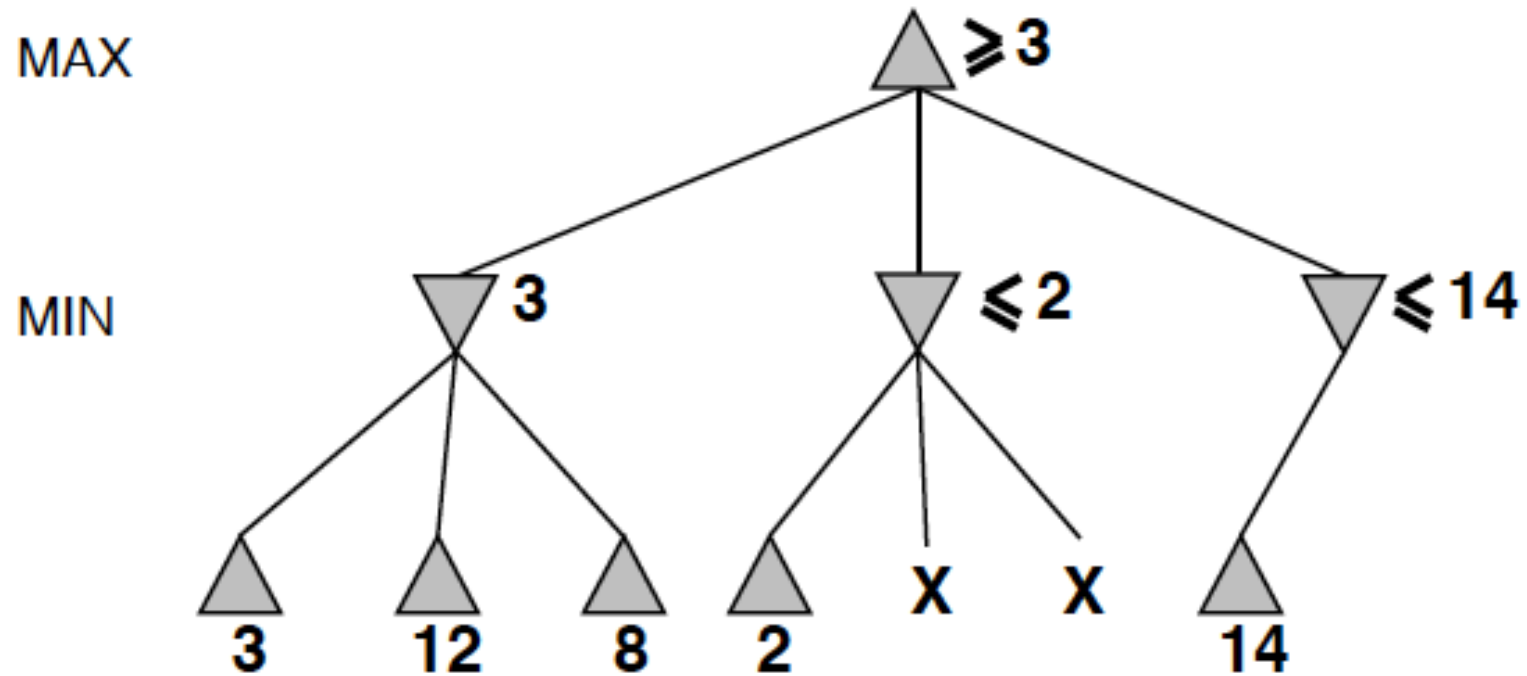
# 알파베타 알고리즘 (1/5)



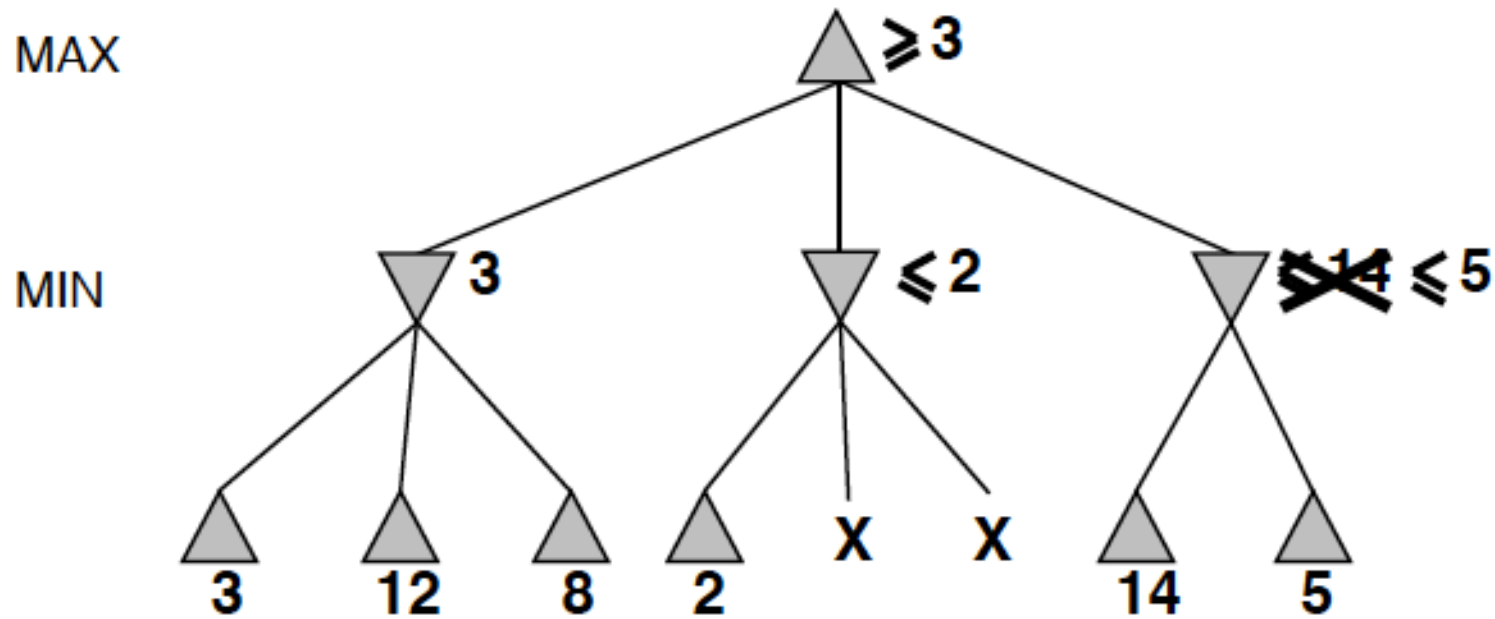
# 알파베타 알고리즘(2/5)



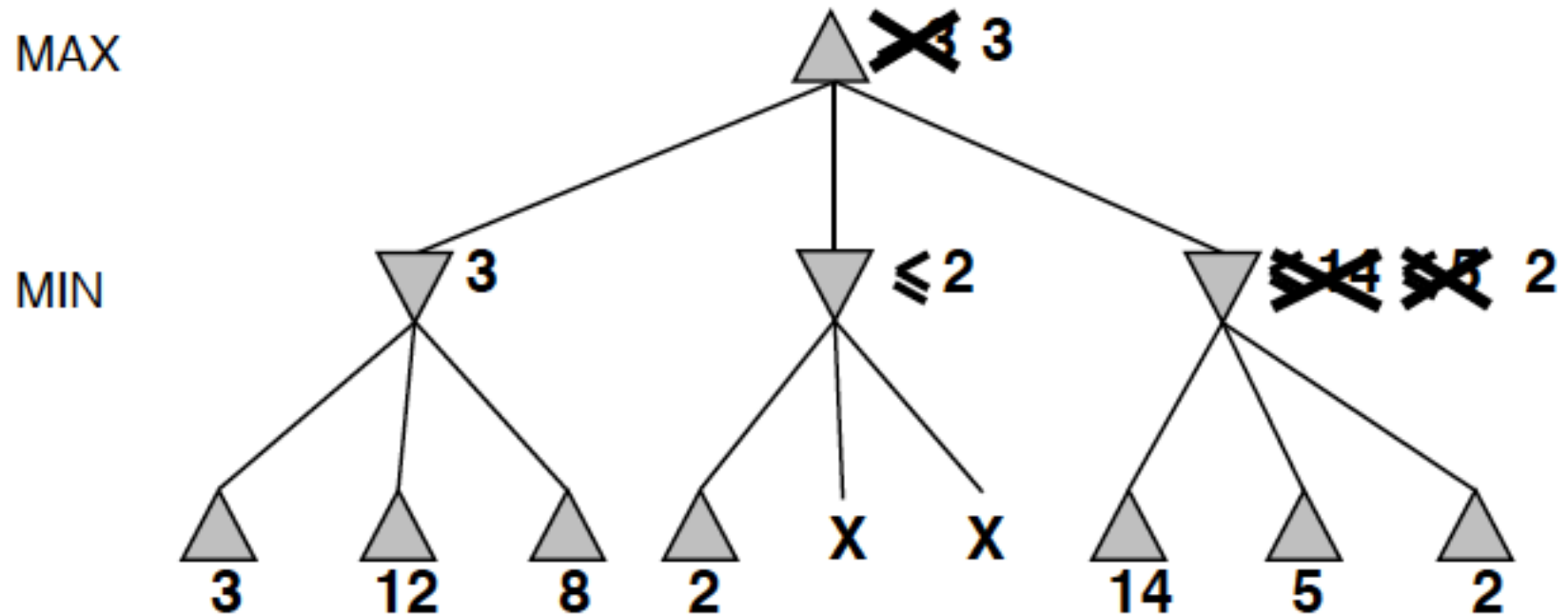
# 알파베타 알고리즘(3/5)



# 알파베타 알고리즘(4/5)



# 알파베타 알고리즘(5/5)



# Reading (Watching) Assignments

- [Silver et al., 2016] Silver, D., ..., Hassabis, D., [Mastering the game of Go with deep neural networks and tree search](#), *Nature* **529**(7587): 484–89, 2016.
  - Q: 바둑 문제의 탐색 공간의 크기는 얼마인가? 복잡도 문제로 인하여 아무리 빠른 컴퓨터를 사용하여도 완전 탐색이 불가능한 바둑 문제에서 알파고가 승률이 높은 수를 빠르게 찾아낼 수 있도록 한 인공지능 기술은 무엇인가?