

# A\* 알고리즘

충남대학교 컴퓨터공학과  
이영석

# A\* 알고리즘

- 최단경로 알고리즘
- 휴리스틱
  - Dijkstra 알고리즘과 유사
  - $f(n) = g(n) + h(n)$ 
    - $g(n)$ : 출발지에서  $n$ 까지 경로 비용
    - $h(n)$ :  $n$ 부터 목적지까지 추정 경로 비용

# A\* : 퍼즐 맞추기 예



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



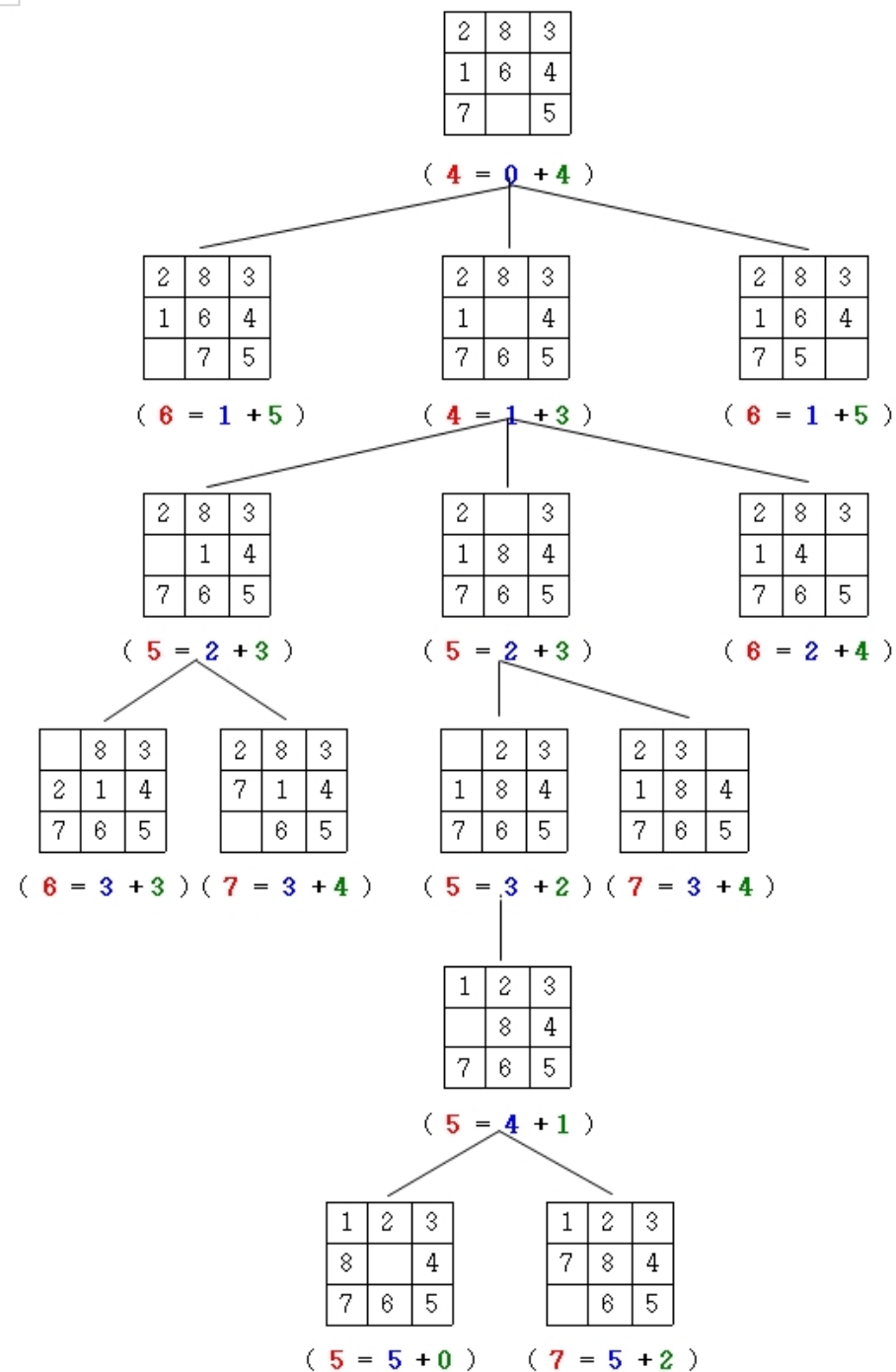
$g(n)$  : 현재까지의 값, 즉 지금까지 움직인 횟수



$h(n)$  : 앞으로 예상되는 값, 위에서는 제자리에 있지 않은 퍼즐의 수

## 알고리즘

- 비용이 최소화되는 경우 선택



```
pq.enqueue(start_node, g(start_node) + h(start_node))    // 우선순위 큐에 시작 노드를 삽입한다.

while pq is not empty    // 우선순위 큐가 비어있지 않은 동안
    node = pq.dequeue    // 우선순위 큐에서 pop한다.

    if node == goal_node    // 만약 해당 노드가 목표 노드이면 반복문을 빠져나온다.
        break

    for next_node in (next_node_begin...next_node_end)    // 해당 노드에서 이동할 수 있는 다음 노드들을 보는 동안
        pq.enqueue(next_node, g(node) + cost + h(next_node)) // 우선순위 큐에 다음 노드를 삽입한다.

return goal_node_dist    // 시작 노드에서 목표 노드까지의 거리를 출력한다.
```

# 예

- 이동가능한 노드에서
  - $f(n) = g(n) + h(n)$  계산
  - $g(n)$ : 출발지에서 경로 비용
  - $h(n)$ : 목적지까지 좌표평면의 거리

- Open List (O)
  - 경로 업데이트

- Closed List (C)
  - 완료

출발지: 0

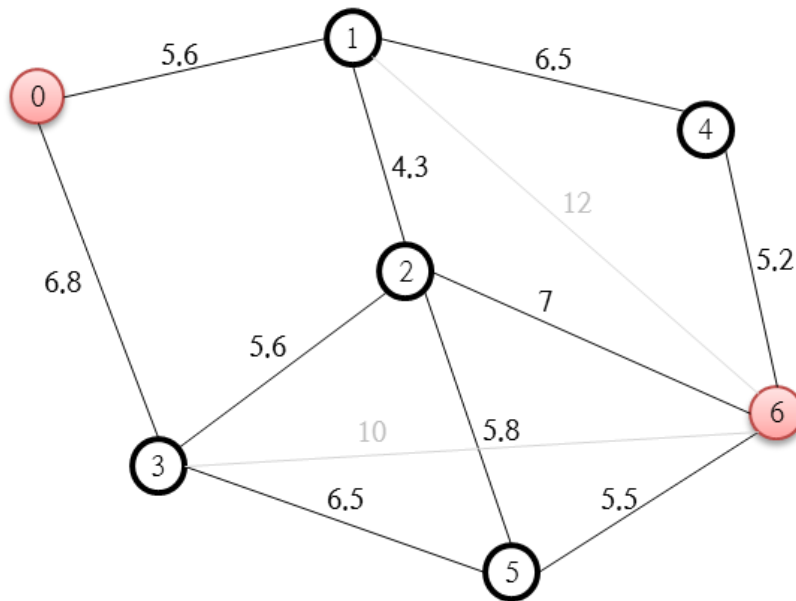
목적지: 6

노드 1의  $H = 12$

- 휴리스틱(추정값): 좌표평면에서 직선거리(Euclidean distance)

노드 3의  $H = 10$

- 휴리스틱(추정값): 좌표평면에서 직선거리(Euclidean distance)



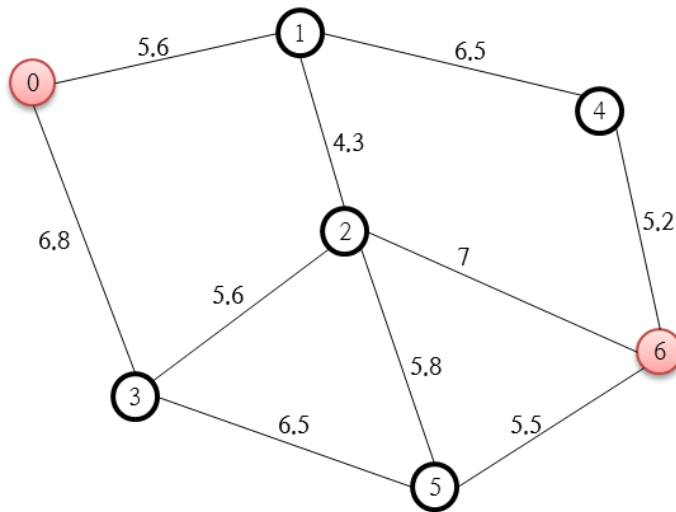
O =

Node ID	1	3
F Score	17.6	16.8
G Score	5.6	6.8
H Score	12	10
Parent Node	0	0

C =

Node ID	0
F Score	0
G Score	0
H Score	0
Parent Node	-

- O 리스트 중 f 최소값 선택
  - 노드 3을 C 리스트에 추가
- 노드 3에서 f, g, h 업데이트
  - 노드 2
    - $g = 6.8 + 5.6 = 12.4$
    - $h = 7$  (추정)
  - 노드 5
    - $g = 6.8 + 5.6 = 13.3$
    - $h = 5.5$  (추정)



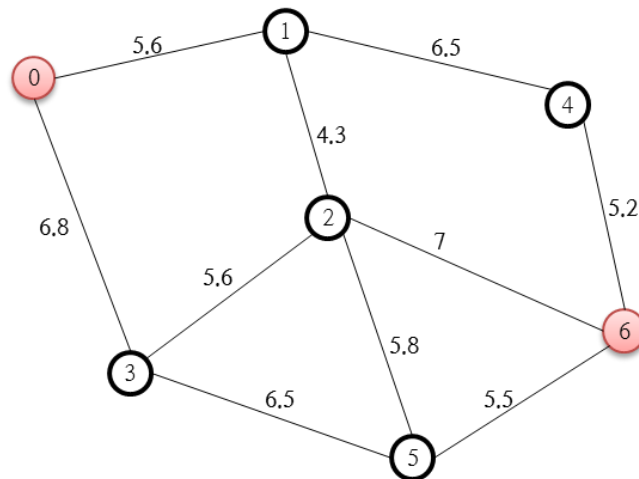
O =

Node ID	1	2	5
F Score	17.6	19.4	18.8
G Score	5.6	12.4	13.3
H Score	12	7	5.5
Parent Node	0	3	3

C =

Node ID	0	3
F Score	0	16.8
G Score	0	6.8
H Score	0	10
Parent Node	-	0

- O 리스트 중 f 최소값 선택
  - 노드 1을 C 리스트에 추가
- 노드 1에서 f, g, h 업데이트
  - 노드 2 -> 이전 값 갱신
    - $g = 5.6 + 4.3 = 9.9$
    - $h = 7$  (추정)
  - 노드 4
    - $g = 5.6 + 6.5 = 12.1$
    - $h = 5.2$  (추정)



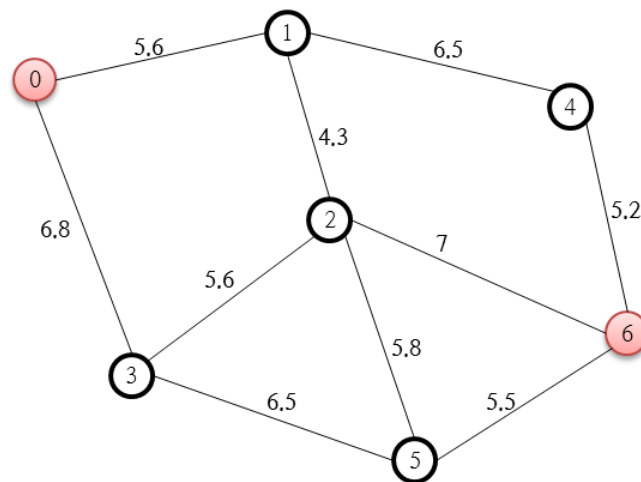
	2		
	19.4		
	12.4		
	7		
	3		
	↓		
Node ID	2	5	4
F Score	16.9	18.8	17.3
G Score	9.9	13.3	12.1
H Score	7	5.5	5.2
Parent Node	1	3	1

O =

Node ID	0	3	1
F Score	0	16.8	17.6
G Score	0	6.8	5.6
H Score	0	10	12
Parent Node	-	0	0

C =

- O 리스트 중 f 최소값 선택
  - 노드 2를 C 리스트에 추가
- 노드 2에서 f, g, h 업데이트
  - 노드 6
    - $g = 9.9 + 7 = 16.9$
    - $h = 0$
  - 노드 5, 4는 그대로



O =

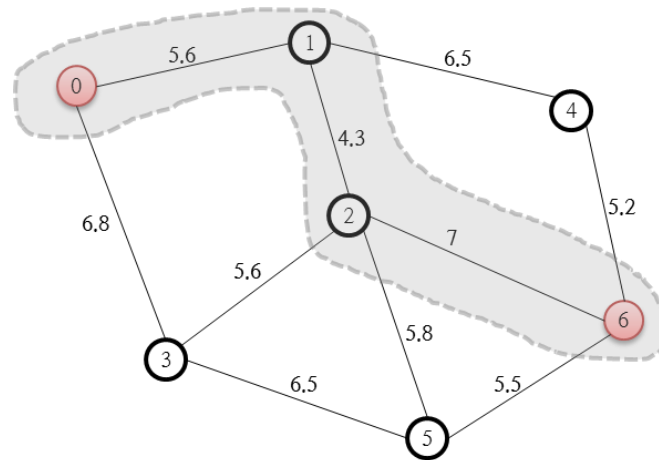
Node ID	5	4	6
F Score	18.8	17.3	16.9
G Score	13.3	12.1	16.9
H Score	5.5	5.2	0
Parent Node	3	1	2

C =

Node ID	0	3	1	2
F Score	0	16.8	17.6	16.9
G Score	0	6.8	5.6	9.9
H Score	0	10	12	7
Parent Node	-	0	0	1



- O 리스트 중 f 최소값 선택
  - 노드 6을 C 리스트에 추가
  - 종료!
- 경로
  - 6 -> 2 -> 1 -> 0



O =

Node ID	5	4
F Score	18.8	17.3
G Score	13.3	12.1
H Score	5.5	5.2
Parent Node	3	1

C =

Node ID	0	3	1	2	6
F Score	0	16.8	17.6	16.9	16.9
G Score	0	6.8	5.6	9.9	16.9
H Score	0	10	12	7	0
Parent Node	-	0	0	1	2



# 최단 경로 알고리즘 비교

- <https://qiao.github.io/PathFinding.js/visual/>