

아래 자료를 기반으로 "Tree Search"를 빠르게 복습·발표할 수 있도록 핵심만 평문으로 정리했습니다. 필요하면 이걸 슬라이드 8~12장으로 바로 쪼갤 수 있습니다.

1) Tree 탐색 한 줄 정의

- 상태공간에서 초기 상태(루트)부터 목표 상태를 찾을 때까지 노드를 확장해 가는 절차. 어떤 노드를 먼저 확장할지에 따라 탐색 전략이 달라진다.

2) 기본 용어와 절차

- 노드 정보: 상태, 부모, 적용 연산자, 깊이, 경로비용 $g(n)$
- 프린지(=프런티어): 확장 대기열. 자료구조가 전략을 결정한다.
 - BFS: 큐, DFS: 스택, UCS/A*/Greedy: 우선순위큐
- 반복 방문 방지: 이미 본 상태는 제외(트리 탐색보다 그래프 탐색이 실용적)

3) 성능 평가 기준

- 완전성(해를 결국 찾는가)
- 최적성(반환 해가 최적해인가)
- 시간복잡도(최악 확장 노드 수), 공간복잡도(최대 보관 노드 수)
- 총비용 = 경로비용(path cost) + 탐색비용(search cost)

4) Uninformed Search 요약

BFS (너비우선)

- 아이디어: 얕은 깊이부터 확장(큐)
- 완전성: O
- 최적성: 단위 간선비용일 때 O
- 시간/공간: $O(b^d)$ 근사
- 용도: 최단 단계 수가 중요한 문제

UCS (균일비용)

- 아이디어: $g(n)$ 이 가장 작은 노드부터 확장(우선순위큐)
- 완전성: O
- 최적성: $O(\text{간선비용 양수}, g \text{가 단조 증가})$
- 시간/공간: $O(b^d)$ 근사
- 용도: 가중치가 있을 때의 최단경로

DFS (깊이우선)

- 아이디어: 현재 경로를 계속 따라 내려감(스택)
- 완전성: X(무한깊이/순환 시)
- 최적성: X
- 시간: $O(b^m)$, 공간: $O(b \cdot m)$
- 용도: 메모리 제약이 큰 때, 빠른 "하나의 해" 시도

깊이제한 탐색(DLS)

- 아이디어: DFS에 깊이 제한 l
- 완전성: $l \geq d$ 이면 O
- 최적성: X
- 시간/공간: $O(b^l)$, $O(b \cdot l)$

반복적 깊이증가(IDS)

- 아이디어: $l=0,1,2,\dots$ 로 DLS 반복
- 완전성/최적성: BFS와 동일(깊이 기준 최적해 O)
- 시간: $O(b^d)$ (DLS 대비 중복 약 10% 내외), 공간: $O(b \cdot d)$
- 장점: BFS의 해 보장 + DFS 수준 메모리

5) Informed Search 핵심

- 공통형식: $f(n)=g(n)+h(n)$
 - $g(n)$: 시작→ n 까지 실제 비용
 - $h(n)$: n →목표까지의 추정 비용(휴리스틱)
- 휴리스틱 조건
 - 허용적(admissible): 실제 비용을 과대평가하지 않음
 - 일관적(consistent): $h(n) \leq c(n,n') + h(n')$ 이면 A^* 가 더 효율적(재방문 감소)

Greedy Best-first

- $f(n)=h(n)$ 만 사용
- 빠를 수 있으나 완전성/최적성 보장 안 됨(사이클·막다른길에 약함)

A^*

- $f(n)=g(n)+h(n)$ 모두 사용
- h 가 허용적이면 최적해 보장, 일관적이면 더 효율적
- 실무 포인트: h 가 강할수록(지배적일수록) 전개 노드 수 감소

6) 최단 경로 예시 포인트(지도 그래프)

- UCS는 누적거리 g 가 작은 노드부터 확장하므로 거리 기준 최적해를 준다.
- Greedy는 직선거리 h 만 보고 진행해 빠르지만 돌아갈 수 있음.
- A^* 는 g 와 h 를 합쳐 균형 있게 확장하며, h 가 허용적이면 최적임.

7) 8-puzzle의 휴리스틱 예

- h1: 제자리에 있지 않은 타일 수
- h2: 타일들의 맨해튼 거리 합
- h2가 h1을 지배(동일하게 허용적일 때 h2가 더 강력 → 노드 전개 수 감소)

8) 알고리즘 비교표(요지)

알고리즘	완전성	최적성	시간	공간	메모
BFS	O	단위비용일 때 O	$O(b^d)$	$O(b^d)$	단계 최단
UCS	O	O	$O(b^d)$	$O(b^d)$	가중치 최단
DFS	X	X	$O(b^m)$	$O(b \cdot m)$	메모리 적음
DLS	$l \geq d$ 면 O	X	$O(b^l)$	$O(b^l)$	깊이 제한 필요
IDS	O	깊이최적 O	$O(b^d)$	$O(b \cdot d)$	BFS≈최적 + 저메모리
Greedy	조건에 따라 X	X	문제 의존	문제 의존	빠른 근사
A*	O	h 허용적이면 O	문제 의존(보통 큼)	큼	h 품질이 핵심

9) 실전 체크리스트

- 그래프 탐색으로 구현하고 방문/폐쇄 집합을 반드시 둔다.
- 비용이 있는 문제에서는 BFS 대신 UCS/A*를 고려한다.
- A*의 h는 “허용적”을 먼저 만족, 가능하면 “일관적”까지.
- 메모리 예산이 작으면 IDS나 IDA* 같은 변형도 고려.

원하면 위 내용을 “발표 스크립트” 버전으로 5~7분, 10~12분 두 길이로 나눠 작성해 드릴게요.