

중간고사 대비 학습 전략 가이드

제공된 강의 자료와 교수님의 암묵적 힌트를 바탕으로, 실전 문제 해결에 초점을 맞춘 학습 프로세스를 제시하겠습니다.



핵심 학습 프로세스 (Extended Thinking 기반)

Phase 1: 문제 이해 및 형식화 (Problem Formalization)

목표: 주어진 문제를 탐색 문제의 구성 요소로 변환

사고 단계:

1. 문제 유형 식별: 8-puzzle? N-queens? 최단 경로?
2. 구성 요소 추출:
 - 초기 상태 (Initial State)
 - 목표 상태 (Goal State) - 명시적/암묵적
 - 연산자 (Operators)
 - 비용 함수 (Cost Function)
3. 제약 조건 파악: 대각선 제약? 무게 제한? Subtour 금지?

검증 포인트:

- ☐ 모든 구성 요소를 명시적으로 나열했는가?
- ☐ 제약 조건을 수식으로 표현할 수 있는가?

Phase 2: 알고리즘 선택 (Algorithm Selection)

목표: 문제 특성에 맞는 최적 탐색 알고리즘 결정

사고 단계:

1. 탐색 유형 판단:
 - 목표까지의 정보 없음 → Uninformed Search
 - 휴리스틱 사용 가능 → Informed Search
2. 성능 요구사항 분석:
 - 최적해 필수? → BFS/UCS/A*
 - 메모리 제약? → DFS/IDS
 - 시간 제약? → Greedy
3. 복잡도 계산: b^d vs $b \cdot m$ 비교

자주 하는 실수:

- ✗ BFS를 선택했는데 "거리 기준 최적해"를 보장한다고 답함
- ✗ DFS의 공간 복잡도를 $O(b^m)$ 으로 씀 (정답: $O(bm)$)
- ✗ A*를 사용했는데 $h(n)$ 이 과대평가되는 휴리스틱을 제시함

Phase 3: 탐색 트리 구성 (Tree Construction)

목표: 노드 확장 순서를 시각화하고 프린지 관리

사과 단계:

1. 루트 노드 생성:

상태: 초기 상태
부모: NULL
깊이: 0
경로 비용: 0

2. 프린지 구조 결정:

- BFS/UCS → Queue (FIFO)
- DFS/DLS/IDS → Stack (LIFO)
- Greedy/A* → Priority Queue ($f(n)$ 기준)

3. 확장 루프:

```
while 프린지 not empty:
    노드 = 프린지에서 선택 (알고리즘 규칙)
    if 목표 상태: return 경로
    자식들 = 노드 확장
    프린지에 자식 추가
```

4. 중복 상태 처리: 이미 방문한 상태 제외

검증 포인트:

- ☐ 각 노드에 (상태, 부모, 깊이, $g(n)$) 표기했는가?
- ☐ 확장 순서 번호를 명확히 기록했는가?
- ☐ 프린지 내용을 각 단계마다 업데이트했는가?

자주 하는 실수:

- ✗ UCS에서 경로 비용을 누적하지 않고 간선 비용만 비교
- ✗ A*에서 $f(n)=g(n)+h(n)$ 계산을 누락하고 $h(n)$ 만 사용
- ✗ IDS에서 각 깊이 제한마다 트리를 다시 그리지 않음

Phase 4: 평가 함수 계산 (Evaluation Function)

목표: Informed Search에서 각 노드의 $f(n)$ 정확히 계산

사고 단계 (A 기준):*

1. **$g(n)$ 계산**: 루트부터 현재까지 실제 경로 비용 합산

$$g(B) = \text{cost}(A \rightarrow B)$$

$$g(C) = \text{cost}(A \rightarrow B) + \text{cost}(B \rightarrow C)$$

2. **$h(n)$ 선택**: 과대평가 금지 원칙

- 최단 경로: 직선 거리 (straight-line distance)
- 8-puzzle: h_1 (잘못된 타일 수) vs h_2 (맨해튼 거리 합)

3. **$f(n)$ 합산**: $f(n) = g(n) + h(n)$

4. **우선순위 정렬**: $f(n)$ 이 작은 노드부터 확장

자주 하는 실수:

- ✗ $h(n)$ 을 실제 최단 거리로 착각 ($h(n)$ 은 추정치!)
- ✗ 맨해튼 거리 계산 시 대각선 이동 포함 (8-puzzle은 상하좌우만 가능)
- ✗ $f(n)$ 이 같을 때 임의로 선택하지 않고 깊이/순서 규칙 무시

검증 포인트:

- ☐ 모든 노드에 $f(n) = g(n) + h(n)$ 표기했는가?
- ☐ 휴리스틱이 admissible(과대평가 없음)한지 확인했는가?

Phase 5: 최적성 검증 (Optimality Verification)

목표: 찾은 해가 정말 최적인지 증명

사고 단계:

1. **알고리즘별 최적성 조건 확인**:

- BFS: 깊이 기준 최적 (간선 비용 동일)
- UCS: 경로 비용 기준 최적 (양의 비용)
- A*: 경로 비용 기준 최적 ($h(n)$ 이 admissible)

2. **반례 탐색**: 더 짧은 경로가 없는지 확인

3. **$f(n)$ 임계값 분석**:

- A*에서 최적해의 비용이 C 라면, $f(n) < C$ 인 모든 노드는 확장됨
- $f(n) \geq C$ 인 노드는 확장되지 않음

자주 하는 실수:

- ✗ "BFS는 항상 최적해"라고 답함 (간선 비용이 다르면 X)

- ✗ Greedy의 해를 최적이라고 주장 (반례 제시 필요)
- ✗ 목표 노드를 처음 발견했을 때 바로 종료 (UCS/A*는 프린지 확인 필요)

Phase 6: 복잡도 분석 (Complexity Analysis)

목표: 시간/공간 복잡도를 정확히 계산하고 비교

사고 단계:

1. 시간 복잡도:
 - 확장된 총 노드 수 계산
 - 분기계수 b , 해의 깊이 d , 최대 깊이 m 파악
 - Big-O 표기: $O(b^d)$, $O(b^m)$, $O(b^l)$
2. 공간 복잡도:
 - 동시에 저장해야 하는 최대 노드 수
 - BFS/UCS: $O(b^d)$ - 모든 프린지 저장
 - DFS/IDS: $O(bm)$ - 한 경로만 저장
3. 실제 값 계산 (예시):
 - $b=10, d=5 \rightarrow$ BFS는 111,111개 노드 확장
 - IDS는 123,456개 (약 10% 오버헤드)

검증 포인트:

- ☐ 표에 있는 복잡도와 실제 계산이 일치하는가?
- ☐ "완전성"과 "최적성"을 혼동하지 않았는가?

자주 하는 실수:

- ✗ DFS의 시간 복잡도를 $O(b^d)$ 로 씀 (정답: $O(b^m)$)
- ✗ IDS의 오버헤드를 무시하고 DLS와 동일하다고 답함
- ✗ 공간 복잡도에서 "총 확장 노드"를 답함 (최대 동시 저장량이 정답)

🌀 실전 문제 유형별 체크리스트

1. 최단 경로 문제

필수 확인 사항:

- ☐ 간선 가중치를 모두 표기했는가?
- ☐ UCS/A* 사용 시 누적 경로 비용 $g(n)$ 정확한가?
- ☐ 직선 거리 휴리스틱이 과대평가하지 않는가?

자주 하는 실수:

- ✗ BFS로 풀고 "거리 최적"이라고 주장
- ✗ A*에서 $h(n)$ 을 실제 거리로 착각

2. 8-puzzle 문제

필수 확인 사항:

- ☐ 초기 상태가 inversion 조건을 만족하는가?
- ☐ 연산자는 공백 타일 기준 상하좌우 이동인가?
- ☐ h1과 h2 중 어떤 휴리스틱을 사용했는가?

자주 하는 실수:

- ✗ 타일 번호를 이동시킨다고 표현 (공백을 이동!)
- ✗ 맨해튼 거리 계산 시 공백 타일 포함
- ✗ 목표 상태와의 inversion 수로 도달 불가능성 판단 누락

3. N-queens 문제

필수 확인 사항:

- ☐ 순열 모델 사용 시 대각선 제약 수식화: $|x_i - x_j| \neq |i - j|$
- ☐ 비용 함수가 "충돌 쌍의 수"인가?
- ☐ 초기해 생성 방법이 명확한가?

자주 하는 실수:

- ✗ 이진 변수 모델과 순열 모델의 경우의 수 차이 무시
- ✗ 대각선 제약을 $i+j, i-j$ 로만 설명하고 절댓값 조건 누락
- ✗ 목표 상태를 "명시적으로 주어진다"고 답함 (암묵적!)

4. 순회 외판원(TSP) 문제

필수 확인 사항:

- ☐ Subtour 제약을 언급했는가?
- ☐ 출발지 고정 시 경우의 수가 $(n-1)!$ 인가?
- ☐ 수학적 모델링에서 x_{ij} 와 c_{ij} 를 구분했는가?

자주 하는 실수:

- ✗ $n!$ 을 $(n-1)!$ 로 줄이지 않음
- ✗ Subtour가 "여러 경로로 쪼개지는 현상"임을 모름
- ✗ 제약 조건에서 "각 도시에서 나가는/들어오는 간선=1" 누락

5. 배낭 문제

필수 확인 사항:

- ☐ 무게(w)와 값(p)을 혼동하지 않았는가?
- ☐ 다차원 배낭에서 여러 무게 제약을 모두 표현했는가?
- ☐ 경우의 수가 2^n 임을 인지했는가?

자주 하는 실수:

- ✗ "값을 최소화"라고 답함 (최대화!)
- ✗ 다차원 배낭에서 하나의 무게 제약만 고려

6. 집합 커버링 vs 최대 커버링

필수 확인 사항:

- ☐ 커버링: 모든 행을 커버, 비용 최소화
- ☐ 최대 커버링: d개 열 선택, 커버된 행 최대화
- ☐ 0-1 행렬의 의미를 이해했는가?

자주 하는 실수:

- ✗ 두 문제의 목적 함수 방향 혼동 (min vs max)
- ✗ 최대 커버링에서 열 개수 제약 d를 누락

7. 탐색 트리 그리기 (교수님 강조!)

필수 확인 사항:

- ☐ 각 노드에 (상태, g(n), h(n), f(n)) 표기
- ☐ 확장 순서를 번호로 명시
- ☐ 프린지 변화 과정을 단계별로 기록
- ☐ 최종 경로를 굵은 선 또는 화살표로 표시

자주 하는 실수:

- ✗ 노드만 그리고 간선 비용 누락
- ✗ 중복 상태를 처리하지 않아 무한 루프 발생
- ✗ A*에서 f(n)이 같을 때 확장 순서 기준 없음

교수님 힌트 재해석:

"탐색 트리 그리는 다른 문제들, 확장 문제들에서 탐색을 잘 그릴 수 있겠죠?"

→ 시험에 탐색 트리 작도 문제가 출제될 가능성 높음! → BFS, UCS, DFS, A* 중 하나를 주고 "깊이 3까지 확장 과정을 그리시오" 유형 예상

최종 점검 체크리스트

이론 암기 사항

- ☐ 4차 산업혁명의 핵심: ICT 융합, 초연결, 초지능
- ☐ ICBM 흐름: IoT → Cloud → Big Data → Mobile
- ☐ 튜링 테스트, 맥카시의 닥터머스 학회 (1956)
- ☐ AI 분류: Think vs Act, Human vs Rationality
- ☐ 약/강 인공지능 차이

- ☐ 지도학습(회귀/이진분류/다중레이블), 비지도학습, 강화학습
- ☐ 전문가 시스템: 지식 베이스 + 추론 엔진

탐색 알고리즘 암기표

알고리즘	완전성	최적성	시간 복잡도	공간 복잡도	비고
BFS	O	깊이 기준 O	$O(b^d)$	$O(b^d)$	큐 사용
UCS	O	경로 비용 O	$O(b^d)$	$O(b^d)$	우선순위 큐
DFS	X	X	$O(b^m)$	$O(bm)$	스택 사용
DLS	조건부	X	$O(b^l)$	$O(b)$	깊이 제한 필요
IDS	O	깊이 기준 O	$O(b^l)$	$O(b)$	약 10% 오버헤드
Greedy	조건부	X	$O(b^m)$	$O(b^m)$	$h(n)$ 만 사용
A*	O	경로 비용 O	$O(b^m)$	$O(b^m)$	$h(n)$ admissible

수식 정리

- 8-queens 대각선 제약: $|x_i - x_j| \neq |i - j|$
- 부대각선: $i + j = \text{constant}$
- 주대각선: $i - j = \text{constant}$
- A* 평가 함수: $f(n) = g(n) + h(n)$
- 8-puzzle h2(맨해튼 거리): $\sum |current_x - goal_x| + |current_y - goal_y|$

마지막 학습 팁

1. 탐색 트리 작도 연습을 최소 5문제 이상 수행하세요.

- 각 알고리즘마다 1문제씩 (BFS, UCS, DFS, A* 필수)
- 시간 재고 손으로 그려보기 (디지털 도구 X)

2. 휴리스틱 설계 문제 대비:

- "직선 거리가 왜 admissible한가?" 설명 가능해야 함
- 8-puzzle에서 h1 vs h2 비교 논리 정리

3. 문제 정의 서술형 대비:

- 8-puzzle, N-queens, TSP 중 하나를 주고 "문제를 정의하시오" 예상
- 초기/목표 상태, 연산자, 비용 함수를 5분 안에 작성 연습

4. 실수 노트 작성:

- 이 문서의 "자주 하는 실수" 항목을 카드로 만들어 시험 직전 복습

이 프로세스를 따라 공부하시면 교수님의 암묵적 힌트와 강의록의 핵심을 모두 커버할 수 있습니다. 특히 **탐색 트리 작도 능력**이 변별력을 가를 것으로 예상되니 집중 연습하세요!

