



# 백트래킹

💡 해가 아니면 되돌아가서 다시 해를 찾아가는 기법

## 백트래킹 (Back Tracking) 기법

⇒ 해가 아니면 되돌아가서 다시 해를 찾아가는 기법

↳ 최적화 (optimization) 문제와 결정 (decision) 문제 해결가능

↳ 최단거리찾기 ... 문제의 조건에 맞는 해가 존재하는지 Yes / No  
최대원 ...  
ex) 미로찾기, n-Queen, Map coloring  
부분집합의 합 문제 - 원소의 계속적

⇒ 선택은 반복하다가 답에 도달하면 완료. 상태공간트리에 대해 탐색 수행

\* 상태공간트리 = 해를 찾기 위한 과정 트리도 표현

↳ 상태공간트리 탐색 → 모든 후보해 탐색

트리를 깊이 우선 탐색 하는 방법 : 백트래킹 알고리즘 기본 형태

## \* 백트래킹 VS DFS

- 정리가 해결책으로 이어지지 않으면 경로 따라가기 X ⇒ 시도횟수 ↓

DFS	백트래킹
• 최종 상태에 가는 중간상태 표현	• 가지치기로 불필요한 경로 조기작단
• 경우의 수가 너무 많으면 (N!) 가지 경우의 수를 개진문제 처리	• 일반적으로 경우의 수가 줄어 처리가능
불가능	최악의 경우 지속 탐색 처리시간 오래 처리불가능

유망성 점검!! < 유망하다 = 해답가능성 O  
유망하지않다 = 해답가능성 X

\* 8-Queens 문제

8x8 체스판, 8개의 Queen

$$\text{후보해의 수 } 64C8 = \frac{64!}{8!(64-8)!} = 4,426,165,368$$

← 약 45억개

실제 해의 수 92개

\* 백트래킹 알고리즘 절차

값이 우선 탐색에 의해 다음 노드 방문

