

# Backtracking

$$X = [0, 1, 2, 3, \dots, k]$$

Back-track(k): # recursive version

→ X[k] 값 정함

if  $k > m$  : → 재귀 종료

해출됨 ⇒ if solution-check(x) : print solution

return

for X[k]이 들어갈 수 있는 모든 c에 대해서

if  $B(x[0], \dots, x[k-1], c) == \text{TRUE}$  : → (x[0], ..., x[k], c)이게 가능한 해인가 (가져오면)

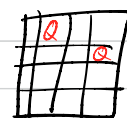
→ candidate

X[k] = c

Backtrack(k+1)

B: Bounding Function (한계 함수)

1) NQueens:



⇒ Q를 둘 수 있는지 여부 확인 함수 (B)

\* Backtrack 알고리즘의 성능

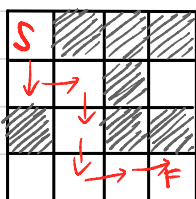
⇒ 한계 함수 B의 효율성

2) Subset Sum:  $A = [2, 5, 7, \dots, c]$  (등차수열)

$$B: \text{sum} + c \leq S$$

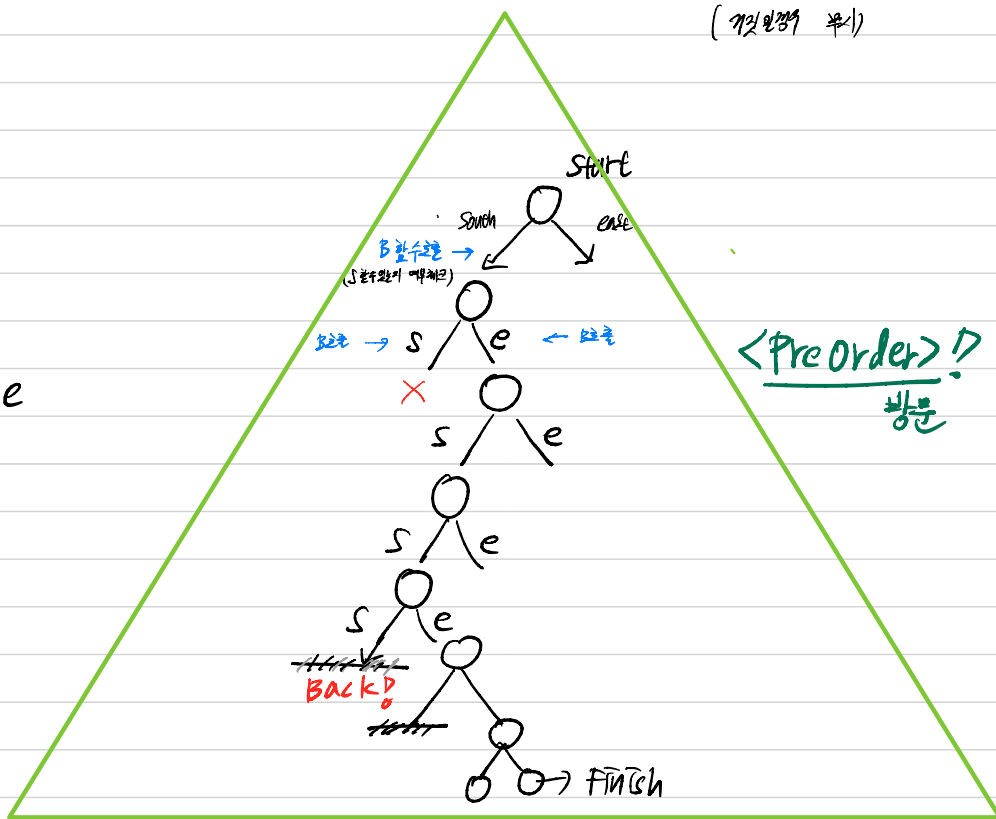
⇒ B가 참일 때까지, X[k]이 c를 지정 (가장 먼저 무시)

\* 퍼즐을 문제



4x4

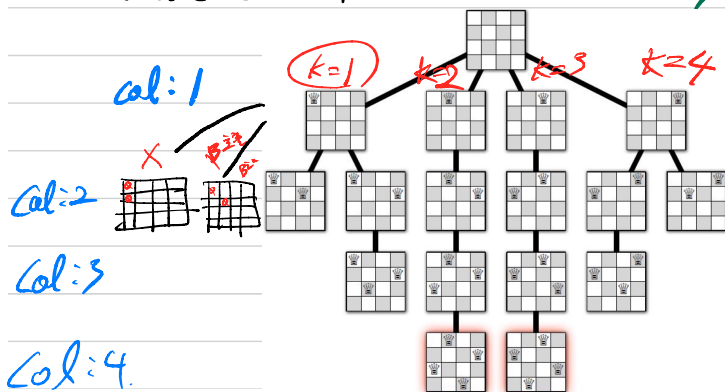
해: S → e → S → S → e → e



<Preorder>? 방문

\* N Queens 문제

⇒ State Space Tree



# \* Subset Sum Problem

$X = [ \underline{1}, \underline{0}, \underline{1}, \underline{0} ]$  2<sup>4</sup>개의 정수 문제

$A = [1, 3, 5, 6]$   $S = 6$

