

# Backtracking

김 태 현

# Backtracking

## 전체 탐색(broute force) 기반 탐색 기법

- but, 조건이 충족되지 않는 상황이면 되돌아간다.

### broute force(브루트 포스)

- 무식하게 풀기, 완전 탐색
- 가능한 모든 경우를 전부 탐색하며 요구조건에 충족되는 결과 취득

## 제약 충족 문제(CSP)를 해결하기 적합하다.

- 주어진 제약 조건을 만족시키는 해를 찾는 문제
- 경우의 수가 많아진다면 고도의 가지치기(pruning)가 필요할 수 있다.
- 순열, 조합 문제의 응용
- 스도쿠, N-퀸

## 상태 공간 트리(state space tree)

- 검색 과정은 트리형태로 나타난다.
- leaf node를 문제의 해로 볼 수 있다.
- 최적해는 leaf node 어딘가에 있다.

# 중복 순열, 순열, 중복 조합, 조합

	순열	조합
순서여부	순서가 있다.	순서가 없다.
위치순서	위치와 순서가 중요하다	위치와 순서가 중요하지 않다
표현방법	배열한다.(선택+배열)	뽑는다.(선택)
동일판단	$\{a, b\} \neq \{b, a\}$	$\{a, b\} = \{b, a\}$
계산방법	${}_nP_r = {}_nC_r \times r!$	${}_nC_r$
배열방법	배열하는 방법이 정해지지 않을 때 	배열하는 방법이 한 가지로 정해져 있을 때 
자격조건	자격조건이 다르다	자격조건이 같다

순열과 조합의 차이를 명확하게 이해했는가!

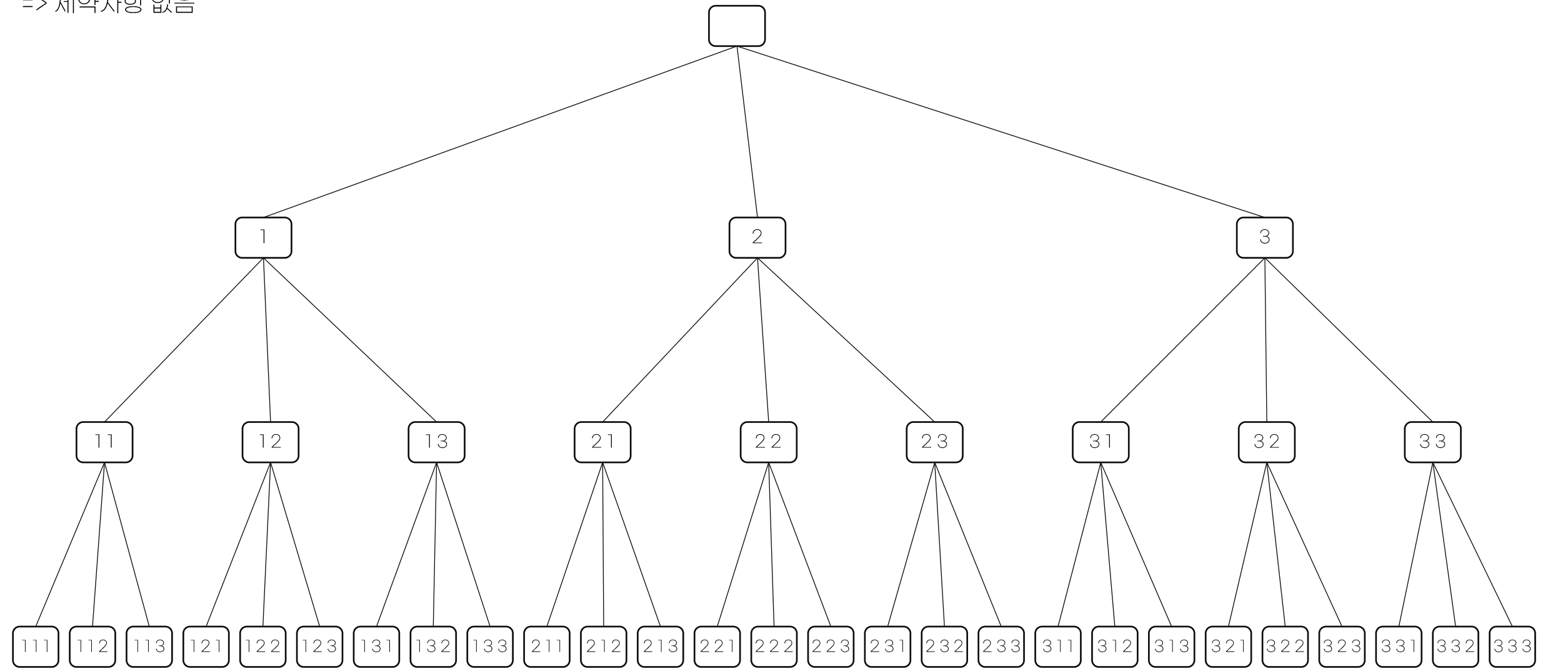
	중복 비허용	중복 허용
순서 고려	순열 P	중복순열 $\Pi$
순서 무시	조합 C	중복조합 H

서로 다른 3개의 문자 A, B, C에서 2개를 택하는 순열, 중복순열, 조합, 중복조합의 차이점은 무엇인가?

	순서	중복	표현	배열 방법
순열	○	×	${}_3P_2$	AB, BA, BC, CB, CA, AC
중복순열	○	○	${}_3\Pi_2$	AB, BA, BC, CB, CA, AC + AA, BB, CC,
조합	×	×	${}_3C_2$	AB, BC, CA
중복조합	×	○	${}_3H_2$	AB, BC, CA + AA, BB, CC

# 중복 순열

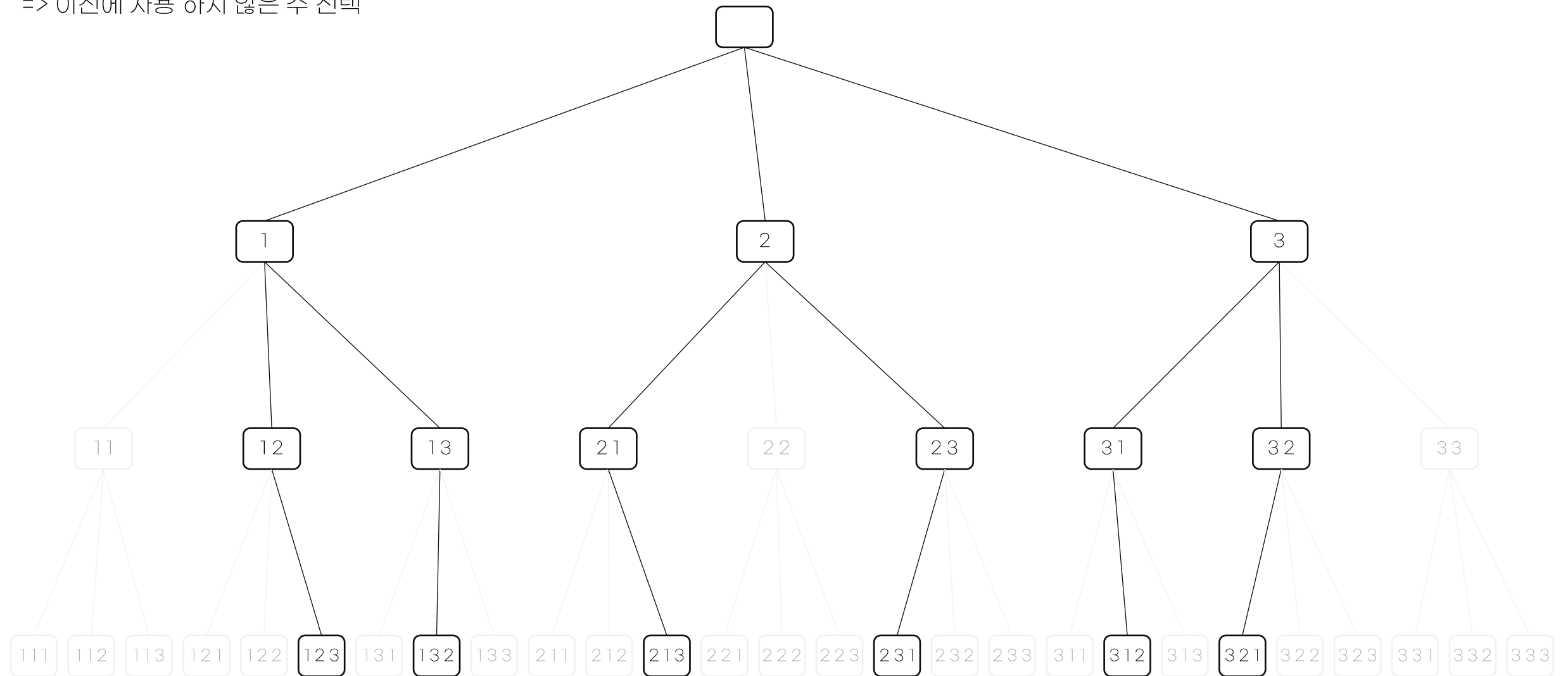
1~3 자연수 중복 **있**이 3개 나열(순서○)  
=> 제약사항 없음



# 순열

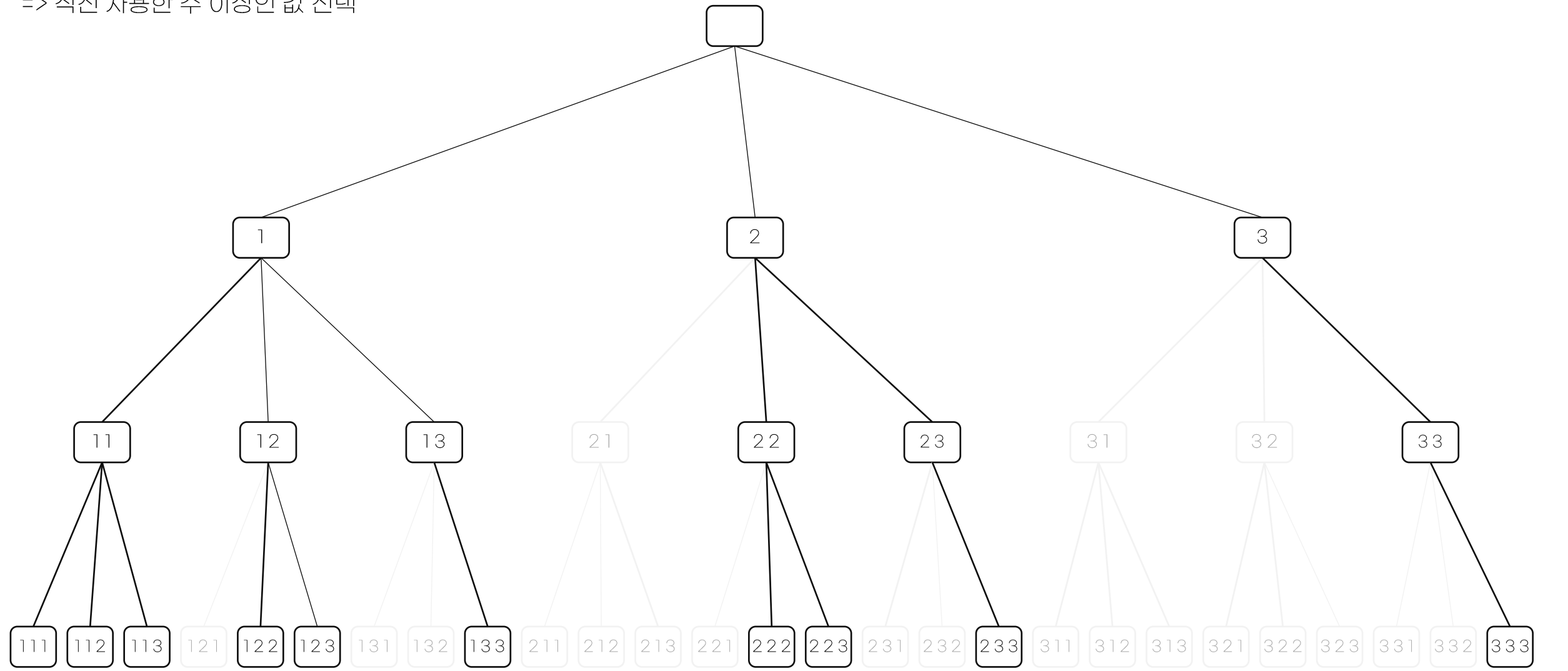
1~3 자연수 중복 **없이** 3개 나열(순서O)

=> 이전에 사용 하지 않은 수 선택



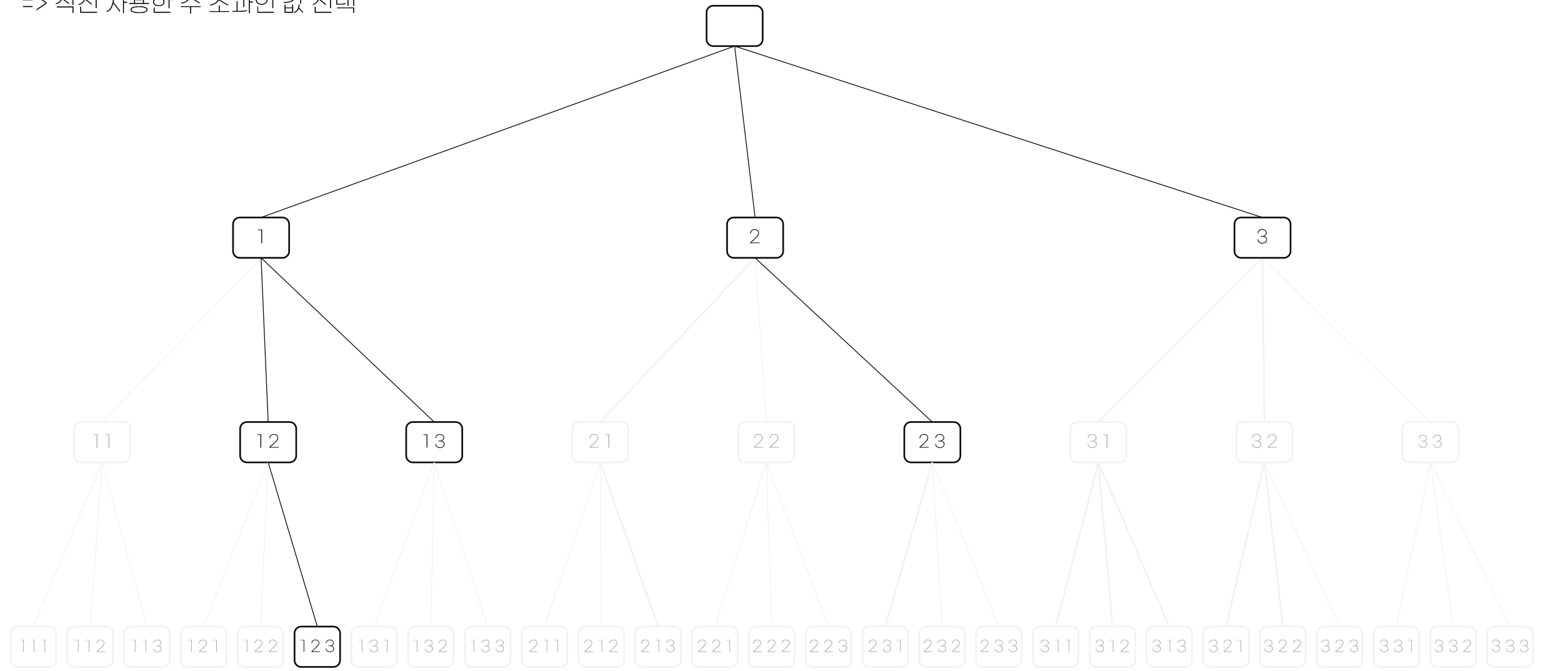
# 중복 조합

1~3 자연수 중복 **있**이 3개 뽑기(순서X)  
=> 직전 사용한 수 이상인 값 선택



# 조합

1~3 자연수 중복 없이 3개 뽑기(순서X)  
=> 직전 사용한 수 초과인 값 선택



**감사합니다**