

COP 3502 11/21/25

Backtracking! vs Brute Force

Difference is very minor!

b.f

Solve (Partial Solution, ^{int* perm} ~~int k~~ Current Slot) ^{↓ int k, other stuff}

if ($k == n$) ^{# item to complete} {
 evaluation
 return
}

for (ALL POSS ITEMS x in SLOT k) {

 # 1. Put x in SLOT k

 2. Solve (Part Sol, k+1)

 3. Do some action

 4. Undo x in SLOT k

}

In
backtracking
JUST ADD
at IF SENT
SKIP OVER
CHOICES DOOMED
TO FAIL!

$k/9 \rightarrow r$
 $k \% 9 \rightarrow c$

```
int  
go(board[9][9], int k) {  
    if (k == 81) unfilled  
        return 1; // true ↓ return  
    if (board[k/9][k%9] != 0) go(board, k+1);  
    for (int i = 1; i <= 9; i++) {  
        // backtrack  
        if (!possible(board, k, i))  
            continue;  
  
        board[k/9][k%9] = i;  
        int res = go(board, k+1);  
        if (res) return 1;  
        board[k/9][k%9] = 0;  
    }  
    return 0;  
}
```

BOX CODE

$k = 0$ to 26

k / SIZE 0, 1, or 2

$\text{Box} \rightarrow 0$

$k = 27$ to 53 $(k / \text{SIZE}) / \text{Box} = 1$

$k = 54$ to 80 $(k / \text{SIZE}) / \text{Box} = 2$

$\text{Box} \in (0, 1, \text{or } 2) \rightarrow \{0, 3, 6\}$

$(k \% \text{SIZE}) / \text{Box}$

↳ for COL