

6/20/2025 Comp Prog 0,1,2,3

oddm (
^{digits}

2

1

 , $k=2$, $nD=4$)

$\left[\begin{array}{l} 2100 \\ 2101 \\ 2102 \\ 2103 \\ 2110 \\ 2111 \\ 2112 \\ 2113 \end{array} \right]$	$\text{oddm}(210-,$ $k=3,$ $nD=4)$	$\left[\begin{array}{l} 2120 \\ 2121 \\ 2122 \\ 2123 \\ 2130 \\ 2131 \\ 2132 \\ 2133 \end{array} \right]$	$\text{oddm}(212-,$ $k=3,$ $nD=4)$
$\left[\begin{array}{l} 2110 \\ 2111 \\ 2112 \\ 2113 \end{array} \right]$	$\text{oddm}(211-,$ $k=3,$ $nD=4)$	$\left[\begin{array}{l} 2130 \\ 2131 \\ 2132 \\ 2133 \end{array} \right]$	$\text{oddm}(213-,$ $k=3,$ $nD=4)$

```

for (int i=0; i<nD; i++) {
    oddmdigits[k] = i; // Try digit i in lock k
    oddm(digits, k+1, nD);
}

```

Move to Permutations

0,1,2	n	choices	1 st item
0,2,1	n-1	"	2 nd item
1,0,2	n-2	"	3 rd item
1,2,0	:		
2,0,1			
2,1,0			
	x 1	choice n th item	
	n!		

odometer allows repeats
perm does NOT!

```
for (int i = 0; i < nD; i++) {  
    // Skip if i has been placed!  
    digits[k] = i
```

```
}  
    :
```

digits

3	0	4			
---	---	---	--	--	--

used

T	F	F	T	T	F
0	1	2	3	4	5

Store
BOTH!

Could loop through
old digits...
OR
store in a boolean
vector which digits
have been used

if (used[i]) continue;

Derangements

Permutation where

$perm[i] \neq i$ for all i .

P 0 1 2

Derange
ment 2 0 1

Brute Force / Backtracking Archetype

if (filled out) {

PROCESS

RETURN

}

LOOP THROUGH ALL POSSIBLE
OPTIONS IN SLOT K {

SKIP OPTION IF DOOMED TO FAIL

PLACE OPTION IN SLOT K
RECURSIVELY SOLVE (arr, k+1)

UNDO MOVE IN SLOT K

}

UPWARDS

a c f h
↑ ↑ ↑
① ② ①

4 letter

1 gap upward

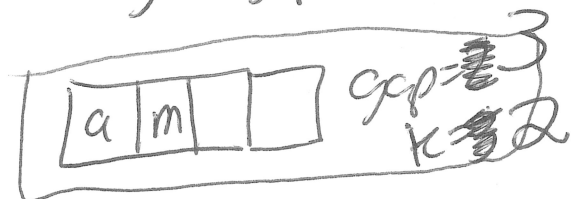
Input n = # letters

g = gap size

n=3

g=4

afk agl
af l ag m
af m ↓
↓ ag z
af z etc.



	0	1	2	3	4	5	6	7
old	0	0	1	<u>1</u>	0	1	1	1
new	0	0	<u>1</u>	2	2	3	4	<u>5</u>

5 non squares
≤ 7.

$$\text{new}[i] = \underline{\underline{\text{new}[i-1]}} + \text{old}[i];$$