

2/2/2022 CSI - Recursion Last Day

Last Time

- ① Odometer
- ② Combinations

Left

- ③ Permutations
- ④ Derangements
- ⑤ Upwards

↓ 0, 1, 2, n-1

0, 1, 2
0, 2, 1
1, 0, 2
1, 2, 0
2, 0, 1
2, 1, 0

<u>LOCS</u>	
0	[2, 1]
1	[-3, 17]
2	[0, 15]

perm represents this path

$(0, 0) \rightarrow (-3, 17) \rightarrow (0, 15) \rightarrow (2, 1) \rightarrow (0, 0)$

dist (LOCS[perm[i+1]],
LOCS[perm[i]])
res +=

perm used k length
 perm([2|4|_|_|], [0|0|0|1], 2, 5)

([2|4|0|1|3]
 [2|4|0|3|1]
 [2|4|1|0|3]
 [2|4|1|3|0]
 [2|4|3|0|1]
 [2|4|3|1|0]

perm([2|4|0|_|_|], [1|0|0|1|0|1], 3, 5)
 perm([2|4|1|_|_|], [0|1|1|1|0|1], 3, 5)
 perm([2|4|3|_|_|], [0|0|1|1|1|1], 3, 5)

Try all values in slot k skipping the used ones.

Derangement

n people wearing n hats.

Want to exchange so no person gets their own hat, how many ways can they do this?

We will print all permutations where value in index $i \neq i$.

Upward

letters are in alpha order no repeats

$9 C m \times 2$

k upward, gap btw consecutive letters
is at least k letters

$k=2$

ad

↘ (b, c not allowed)

Goal: Print all k Upwards of length n

printUp (char* word, int slot, ^{int skip}int n)

```
if (slot == n) {  
    printf("%s", word);  
    return;  
}
```

}

char start = 'a';

if (slot > 0) start = word[slot-1] + skip + 1;

```
for (char next = start; next <= 'z'; next++) {  
    word[slot] = next;  
    printUp(word, slot+1, skip, n);  
}
```

}