COP3502 9/26/23

fyi - I have to leave right after class!

1. P2 - posted, solution
2. Rec RP2 - look @ last week's notes
3. LAST Recursion Lec - Brute Force
   - a) ODOMETER - I made up
   - b) COMBOS } well known
   - c) PERMUTATIONS } well known
   - d) DERANGEMENTS } well known
   - e) UPWARDS - I made up

numD = 5

n = 4

0 0 0 0
0 0 0 1
0 0 0 2
0 0 0 3
0 0 0 4
0 0 1 0
⋮
0 0 4 4
0 1 0 0
⋮
4 4 4 4

odometer([2|1|], 2, 4) — k = 2, n = 4 (# fixed slots, # total slot)

odometer([2|1|0], 3, 4) → 2100, 2101, ⋮, 2104

odometer([2|1|1], 3, 4) → 2110, ⋮

odometer([2|1|2], 3, 4)

2144

odometer([2|1|3])

odometer([2|1|4])

```
for (i = 0; i < NUMDIGITS; i++) {
    odom[k] = i; // fill slot k with digit i
    odometer(odom, k+1, n);
}
```

we always try each value →

Combos

{ mtms, snickers, skittles }

{ }	0	0	0
{ skittles }	0	0	1
{ snickers }	0	1	0
{ snickers skittles }	0	1	1
{ mtms }	1	0	0
{ mtms skittles }	1	0	1
{ mtms snickers }	1	1	0
{ mtms snickers skittles }	1	1	1

Permutations

0, 1, 2, ... , n-1

n=3

0, 1, 2
0, 2, 1

1, 0, 2
1, 2, 0

2, 0, 1
2, 1, 0

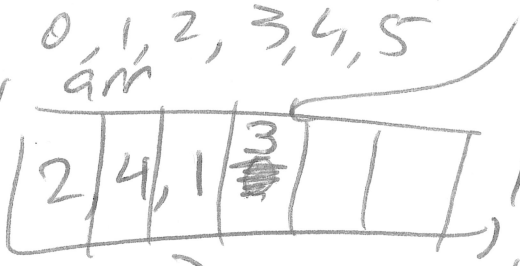
241035
241053

241305
241350

241503
241530

Imagine perm

0, 1, 2, 3, 4, 5




```
for (i = 0; i < n; i++) {
  if (used[i]) continue;
  arr[k] = i; used[i] = 1;
  perm(arr, used, k+1, n);
  used[i] = 0;
}
```

Derangements

0 1 2

1 2 0
2 0 1

In reg perm

if (used[i]) continue;
if (k == i) continue;

a person getting their own hat!!!

Upwards

~~words~~ string letters in sorted order.

a b c d e
a m q z

etc

k-upwards as gap btw letters is at least k letters

a c b d e f 1 upward

a j z 8 upward

n = 3 k = 5

agm agz
agn ahn
ago aho
agp etc.

Problem print out every (k) upward of length (n).

Jump Problem

up ≤ 15

down ≤ 30

Input

n (4)

10 25 8 32

u(15) d(30)