

6/22/2020 - Finish Brute Force

Wednesday, June 24, 2020 10:03 PM

1. Derangements
2. Upwards

Derangements are permutations where the i th value doesn't equal i .

The origination of the problem is something like this:

n people come to a party with hats on. They trade hats so that no person receives their own hat. How many ways can we do this?

$n = 1 \rightarrow 0$

$n = 2 \rightarrow 1$ [1, 0]

$n = 3 \rightarrow 2$ [1, 2, 0], [2, 0, 1]

$n = 4 \rightarrow 9$ (won't list all here is one [2, 3, 1, 0])

First, let's look at the permutation code:

```
// Prints all permutations of perm where the first k items are fixed.  
// perm is of length n, used[i] = 1 iff value I is one of the k fixed items.  
void printPerms(int* perm, int* used, int k, int n) {
```

```
    if (k == n) {  
        print(perm, n);  
        return;  
    }
```

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

```
}  
}
```

What do we have to change, so that we don't ever put value i in slot i?

Well, we already use the if statement to SKIP over stuff we don't want to do, why not skip over more stuff?

CHANGE THE if as follows (now we skip trying to place item k in slot k.)

```
if (!used[i] && i != k)
```

Upwards: An Up-word is an arrangement of letters that are in strictly increasing alphabetical order. For example, "act", "best" are both up words. Now, we define a k skip upword where the letters are in increasing alpha order, and we SKIP at least k letters in between consecutive letters. So, if k = 2 and our first letter is 'a', the both b and c can't be used, 'd' is the next allowable letter.

"adiq" would be a valid 2-skip upword that is 4 letters long.

Input would be k, how many letters must be skipped, and n, the length of the upward.

Problem: Print all k skip upwards of length n in alphabetical order.

3 letter upwards skip of 2

```
adg  
adh  
adi  
...  
adz  
aeh  
aei  
...  
aez  
afi
```

