Counting permutations $\varphi : [n] \longrightarrow [n]$
with constraints: forced or forbidden
outputs.

1. use factorial to count any unconstrained permutations, after constraints met
2. all perm's are functions, so cannot have two outputs for one input at same time: Cannot have $\varphi(j) = i$ AND $\varphi(j) = k$.
3. all perm's are one-to-one, (injections) so cannot have 2 inputs going to same output: Cannot have $\varphi(j) = i$ AND $\varphi(k) = i$, simultaneously.

→ For multiple constraints, keep track with a tree.

Ex: Count perms $\varphi : [5] \to [5]$ such that

$$\varphi(1) \neq 3,\ \varphi(1) \neq 5,\ \varphi(2) \neq 3,\ \varphi(2) \neq 4,\ \varphi(5) \neq 4$$

Ans:

$$5! - 5(4!) + 6(3!) - 1(2!) + 0 - 0$$

total

$5-1=4$

$5-2=3$

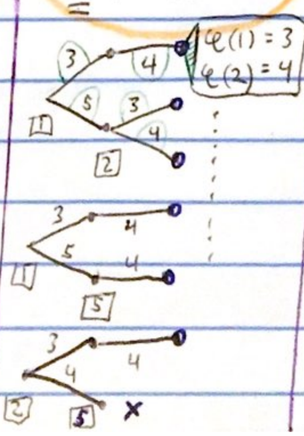
$5-3=2$

No way to break **four** rules.

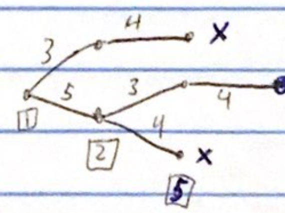
Five separate rules, each fixing one output

factorial on remaining outputs

Six possible ways to break 2 rules:



One way to break 3 rules at a time.



$\boxed{= 34}$