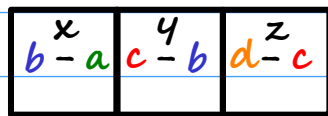
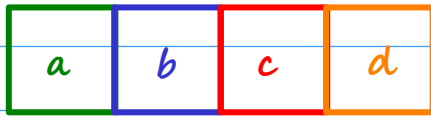


Before the whole $O(N)$ shenanigans...

we are going to do some dumb things with differences...

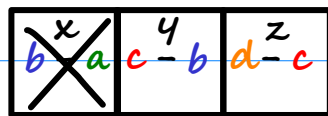
1) Transform into differences ($b - a$ OR $a - b$ does not matter...)



2) How to delete numbers if we have only differences?

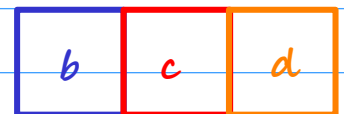
2A) How to delete the first number?

Drop a



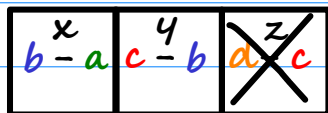
$y : z : []$

aka:



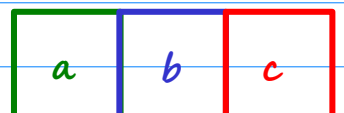
2B) How to delete the last number?

Drop d



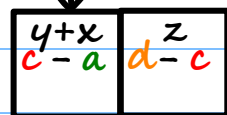
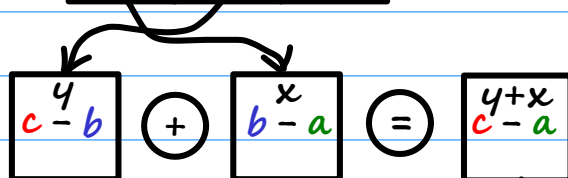
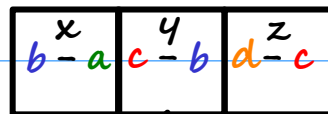
$x : y : []$

aka:



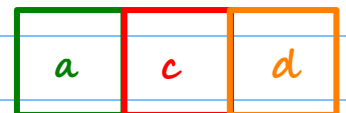
2C) How to delete the other numbers?

Drop b

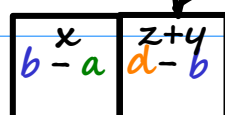
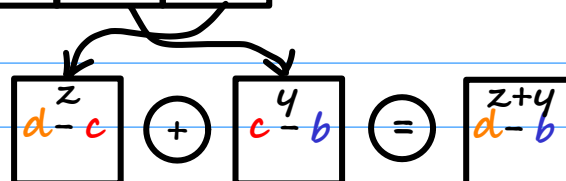
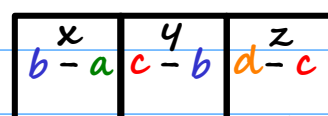


$(y + x) : z : []$

aka:

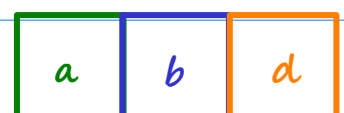


Drop c



$x : (z + y) : []$

aka:



So... $O(N * N)$ is slow, we want the juicy $O(N)$...

Let's see the Maneater variant

1) How to represent state?

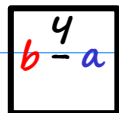
$([Int], Int, [Int])$
all OK differences \swarrow \uparrow \nwarrow remaining differences
can be a Maybe
we need just 1 look-back current difference

2) Cases

2A) We are at the last one

$([], y, [])$

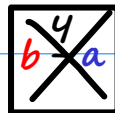
Let's say y is ok \rightarrow we can just keep (or drop last) \rightarrow True



Let's say y is bad

We can drop $a \rightarrow$ True

We can drop $b \rightarrow$ True

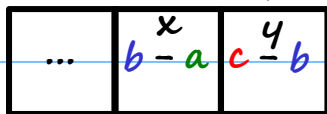


2B) We are at the last one

$(x:_, y, [])$

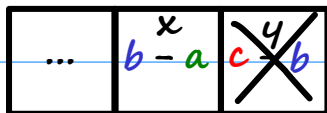
Let's say y is ok

we can just keep it \rightarrow True



Let's say y is bad

We can drop $c \rightarrow$ True



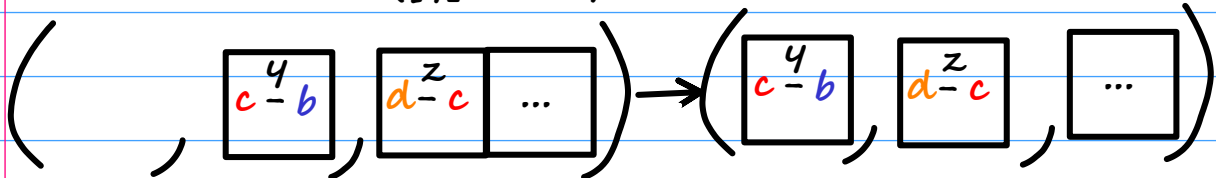
Maneater continued...

2C) We are at the first one

$([], y, z:zs)$

Let's say y is ok

we recurse to $([y], z, zs)$



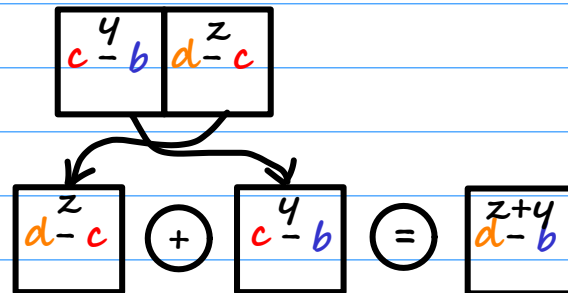
Let's say y is bad

We can drop b , but from now on we cannot delete more, so

Drop b all $\begin{bmatrix} \cancel{c^y} & \cancel{b} & d^z - c & \dots \end{bmatrix}$ must be ok

We can drop c , but from now on we cannot delete more, so

Drop c



all $\begin{bmatrix} d^{z+y} - b & \dots \end{bmatrix}$ must be ok

OR $\begin{bmatrix} d^z - c & \dots \end{bmatrix}$ AND all $\begin{bmatrix} \dots \end{bmatrix}$ must be ok

Diagram showing the final result. The boxes $[d^z - c]$ and $[d^{z+y} - b]$ are shown in a large bracket, indicating they are the final result. The text "OR" is written between them. The text "AND all $\begin{bmatrix} \dots \end{bmatrix}$ must be ok" is written to the right.

Maneater continued...

2D) We are at an inside one

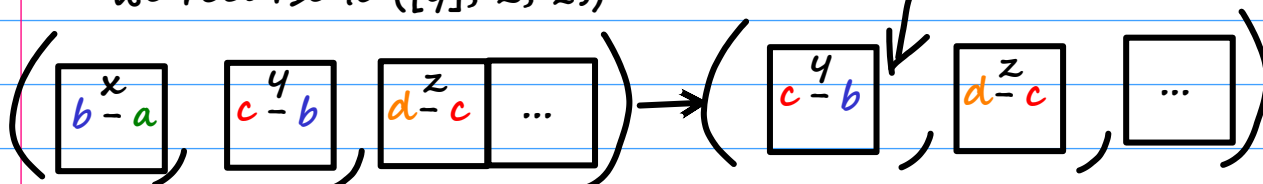
$(x:_, y, z:zs)$

Let's say y is ok

we recurse to $([y], z, zs)$

You can forget 'x'

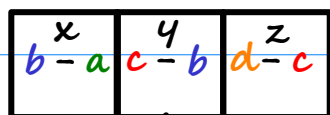
it will never be used again



Let's say y is bad

We can drop b , but from now on we cannot delete more, so

Drop b

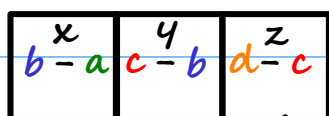


$$\begin{array}{|c|} \hline y \\ \hline c - b \\ \hline \end{array} + \begin{array}{|c|} \hline x \\ \hline b - a \\ \hline \end{array} = \begin{array}{|c|} \hline y+x \\ \hline c - a \\ \hline \end{array}$$

all $\begin{array}{|c|} \hline y+x \\ \hline c - a \\ \hline \end{array} \begin{array}{|c|} \hline z \\ \hline d - c \\ \hline \end{array} \dots$ must be ok

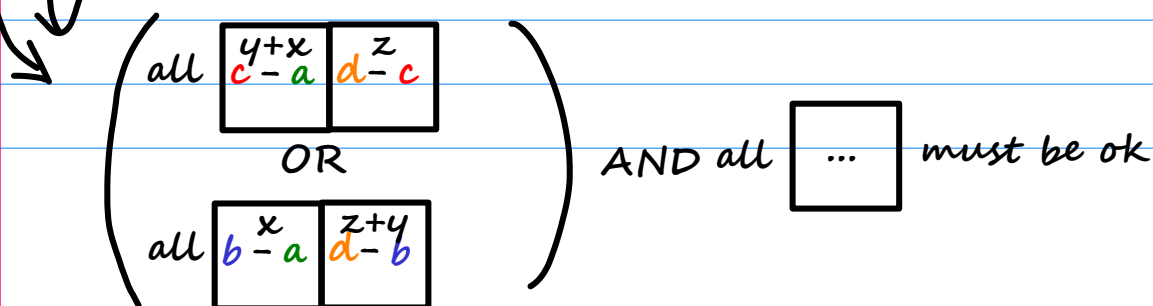
We can drop c , but from now on we cannot delete more, so

Drop c



$$\begin{array}{|c|} \hline z \\ \hline d - c \\ \hline \end{array} + \begin{array}{|c|} \hline y \\ \hline c - b \\ \hline \end{array} = \begin{array}{|c|} \hline z+y \\ \hline d - b \\ \hline \end{array}$$

all $\begin{array}{|c|} \hline x \\ \hline b - a \\ \hline \end{array} \begin{array}{|c|} \hline z+y \\ \hline d - b \\ \hline \end{array} \dots$ must be ok



Maneater Summary

3) Maneater works only with non-empty lists

So... nest Maneater inside a charade function

on empty list \rightarrow return True

else \rightarrow call Maneater

Summary:

We DO NOT care about anything else, we just want $O(N)$ like for (d : differences)

if(!p(d)) return false

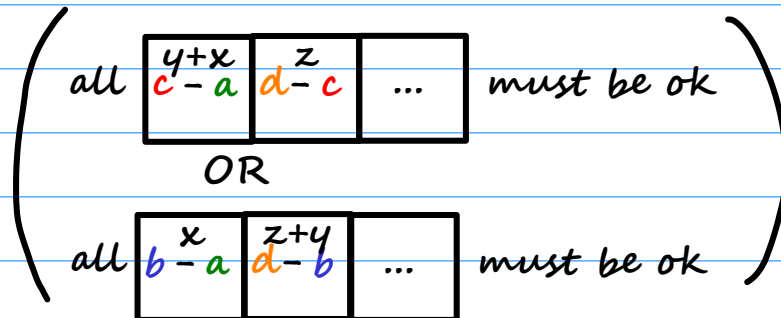
Unfortunately, we cannot do it immediately

So we play nice... bide our time...

BUT as soon as we 'use up' our delete choice

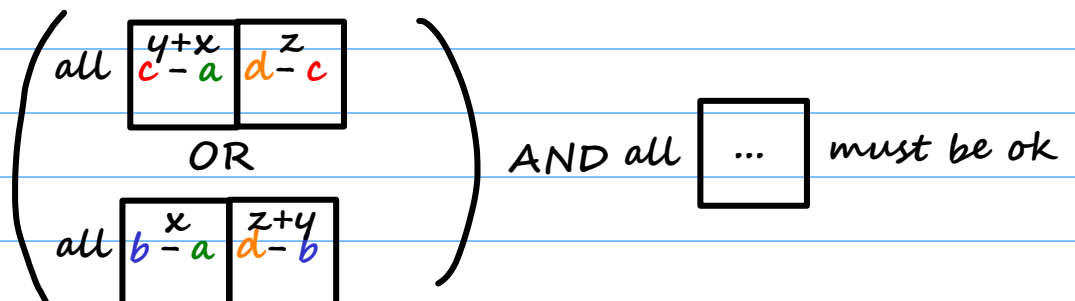
we show our true colors and TURN INTO A FOR-LOOP

But we must be smart... We don't want to do 2 loops, like:



So only if all preemptive checks are fine,

do we turn into a for-loop



Runtime characteristics

Below is a video detailing runtime characteristics:

<https://youtu.be/KErUJUzUpO?t=122>