Updating the Problem

OPT(i,j,f) is the maximum amount of eggs Baby Yoda can collect on a legal path from (i,j) to (0,0) using the force f times to knock over rocks.

For simplicity, assume there are no rocks at the starting location (r-1,c-1)

Here was the old rule without the force – how do we update?

$$OPT(i,j) = \begin{cases} -\infty & \text{if } rocks(i,j) \text{ is true} \\ -\infty & \text{if } i < 0 \text{ or } j < 0 \end{cases}$$

$$eggs(0,0) & \text{if } i = 0 \text{ and } j = 0$$

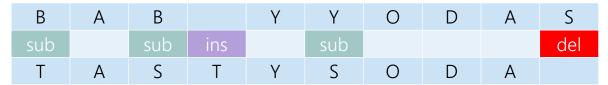
$$max\{OPT(i-1,j), OPT(i,j-1)\} + eggs(i,j) & \text{otherwise} \end{cases}$$

Dynamic Programming Process

- 1. Define the object you're looking for
- 2. Write a recurrence to say how to find it
- 3. Design a memoization structure
- 4. Write an iterative algorithm

Example

What's the distance between babyyodas and tastysoda?



Distance: 5, one point for each colored box

Quick Checks – can you explain these?

If x has length n and y has length m, the edit distance is at most $\max(x, y)$

The distance from x to y is the same as from y to x (i.e. transforming x to y and y to x are the same)

Edit Distance Fill in the next two entries. Be careful with the sub/match distinction!											
OPT	(i,j)	0	В, 1	A, 2	В, 3	Y, 4	Y, 5	O, 6	D, 7	A, 8	S, 9
0		0	1	2	3	4	5	6	7	8	9
Т	1	1	1	2	3	4	5	6	7	8	9
Α	2	2	2	1	2	3	4	5	6	7	8
S	3	3	3	2	2	3	4	5	6	7	7
Т	4	4	4	3	3	3	4	5	6	7	8
Υ	5	5	5	4	4	3					
S	6										
0	7										
D	8										
Α	9										