CSCI 232:
Data Structures and Algorithms

Dynamic Programming (Part 2)

Reese Pearsall
Spring 2025

Edit Distance Problem

Given two strings, how many edits are needed to turn one string into the other?

String 1: **SNOWY**
String 2: **SUNNY**

Edit Distance Problem

Given two strings, how many edits are needed to turn one string into the other?

String 1: **SNOWY**
String 2: **SUNNY**

Another way to state this problem:
What is the minimum cost of aligning these two strings?

S N O W Y
S U N N Y

Edit Distance Problem

Given two strings, how many edits are needed to turn one string into the other?

String 1: **SNOWY**
String 2: **SUNNY**

Another way to state this problem:
What is the minimum cost of aligning these two strings?

S	**N**	**O**	**W**	**Y**
S	**U**	**N**	**N**	**Y**
✓	×	✗	×	✓

Cost to align these strings: 3

Edit Distance Problem

Given two strings, how many edits are needed to turn one string into the other?

String 1: **SNOWY**
String 2: **SUNNY**

(It's possible these strings could be different lengths)

Another way to state this problem:
What is the minimum cost of aligning these two strings?

We are allowed to insert "spaces" into the strings

S	**N**	**O**	**W**	**Y**
S	**U**	**N**	**N**	**Y**
✓	✗	✗	✗	✓

S	**-**	**N**	**O**	**W**	**Y**
S	**U**	**N**	**N**	**-**	**Y**

Cost to align these strings: 3

Edit Distance Problem

Given two strings, how many edits are needed to turn one string into the other?

String 1: **SNOWY**
String 2: **SUNNY**

(It's possible these strings could be different lengths)

Another way to state this problem:
What is the minimum cost of aligning these two strings?

We are allowed to insert "spaces" into the strings

```
S N O W Y
S U N N Y
✓ ✗ ✗ ✗ ✓
```

Cost to align these strings: 3

```
S - N O W Y
S U N N - Y
✓ ✗ ✓ ✗ ✗ ✓
```

Cost to align these strings: 3

Edit Distance Problem

Given two strings, how many edits are needed to turn one string into the other?

String 1: **SNOWY**
String 2: **SUNNY**

(It's possible these strings could be different lengths)

Another way to state this problem:
What is the minimum cost of aligning these two strings?

We are allowed to insert "spaces" into the strings

```
S N O W Y
S U N N Y
✓ ✗ ✗ ✗ ✓
```

Cost to align these strings: 3

```
S - N O W Y
S U N N - Y
✓ ✗ ✓ ✗ ✗ ✓
```

Cost to align these strings: 3

```
- S N O W - Y
S U N - - N Y
✗ ✗ ✓ ✗ ✗ ✗ ✓
```

Cost to align these strings: 5

Edit Distance Problem

We want to align two strings $x = [x_1, \dots, x_n]$ and $y = [y_1, \dots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x}_1, \dots, \mathbf{x}_i]$ and $[\mathbf{y}_1, \dots, \mathbf{y}_j]$.

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1, \ldots, x_i}]$ and $[\mathbf{y_1, \ldots, y_j}]$.

Suppose this is the optimal way to align X and Y

S	-	N	O	W	Y
S	U	N	N	-	Y
✓	✗	✓	✗	✗	✓

Cost to align these strings: 3

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1, \ldots, x_i}]$ and $[\mathbf{y_1, \ldots, y_j}]$.

Suppose this is the optimal way to align X and Y

S	-	N	O	W	Y
S	U	N	N	-	Y
✓	✗	✓	✗	✗	✓

Cost to align these strings: 3

-	N	O	W	Y
U	N	N	-	Y
✗	✓	✗	✗	✓

then there is not a more optimal way to align NOWY and UNNY in less than 3 edits (otherwise we would have a more optimal way to align SNOWY and SUNNY

Edit Distance Problem

We want to align two strings $x = [x_1, \dots, x_n]$ and $y = [y_1, \dots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1, \dots, x_i}]$ and $[\mathbf{y_1, \dots, y_j}]$.

Optimal alignments end in one of three ways:

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1}, \ldots, \mathbf{x_i}]$ and $[\mathbf{y_1}, \ldots, \mathbf{y_j}]$.

Optimal alignments end in one of three ways:

$\mathbf{x_i}$

-

Cost: 1

We align a character i from X with a space from Y

S U N N Y
S N O W Y -

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1, \ldots, x_i}]$ and $[\mathbf{y_1, \ldots, y_j}]$.

Optimal alignments end in one of three ways:

	$\mathbf{x_i}$	$\mathbf{-}$
	$\mathbf{-}$	$\mathbf{y_j}$
Cost:	1	1

We align a character j from Y with a space from X

S U N N Y -
S N O W   Y

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1, \ldots, x_i}]$ and $[\mathbf{y_1, \ldots, y_j}]$.

Optimal alignments end in one of three ways:

	$\mathbf{x_i}$	$\mathbf{-}$	$\mathbf{x_i}$
	$\mathbf{-}$	$\mathbf{y_j}$	$\mathbf{y_j}$
Cost:	1	1	{0,1}

We align a character i from X with character j from Y

S U N N Y
S N O W Y

Edit Distance Problem

We want to align two strings $x = [x_1, \dots, x_n]$ and $y = [y_1, \dots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1, \dots, x_i}]$ and $[\mathbf{y_1, \dots, y_j}]$.

Optimal alignments end in one of three ways:

	$\mathbf{x_i}$	**-**	$\mathbf{x_i}$
	-	$\mathbf{y_j}$	$\mathbf{y_j}$
Cost:	1	1	{0,1}

Now must compute $\mathbf{E(i\text{-}1, j\text{-}1)}$

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[x_1, \ldots, x_i]$ and $[y_1, \ldots, y_j]$.

Optimal alignments end in one of three ways:

	x_i	$-$	x_i
	$-$	y_j	y_j
Cost:	1	1	{0,1}

Now must compute $E(i-1, j-1)$

Now must compute $E(i, j-1)$

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$\mathrm{E}(i, j) =$ optimal cost of aligning $[\mathbf{x_1, \ldots, x_i}]$ and $[\mathbf{y_1, \ldots, y_j}]$.

Optimal alignments end in one of three ways:

	$\mathbf{x_i}$	**-**	$\mathbf{x_i}$
	-	$\mathbf{y_j}$	$\mathbf{y_j}$
Cost:	1	1	{0,1}

Now must compute $\mathbf{E(i\text{-}1, j\text{-}1)}$

Now must compute $\mathbf{E(i, j\text{-}1)}$

Now must compute $\mathbf{E(i\text{-}1, j)}$

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j) =$ optimal cost of aligning $[\mathbf{x_1}, \ldots, \mathbf{x_i}]$ and $[\mathbf{y_1}, \ldots, \mathbf{y_j}]$.

Optimal alignments end in one of three ways:

	$\mathbf{x_i}$	–	$\mathbf{x_i}$
	–	$\mathbf{y_j}$	$\mathbf{y_j}$
Cost:	1	1	{0,1}

Now must compute $\mathbf{E(i\text{-}1, j\text{-}1)}$

Now must compute $\mathbf{E(i, j\text{-}1)}$

Now must compute $\mathbf{E(i\text{-}1, j)}$

Select the minimum

Edit Distance Problem

We want to align two strings $x = [x_1, \ldots, x_n]$ and $y = [y_1, \ldots, y_m]$.

$E(i, j)$ = optimal cost of aligning $[\mathbf{x_1, \ldots, x_i}]$ and $[\mathbf{y_1, \ldots, y_j}]$.

x_i	$-$	x_i
$-$	y_j	y_j

$$E(i,j) = \min \begin{cases} E(i-1,j) + 1 \\ E(i,j-1) + 1 \\ E(i-1,j-1) + \text{diff}(i,j) \end{cases}$$

$$\text{where diff}(i,j) = \begin{cases} 0, & x_i = y_j \\ 1, & x_i \neq y_j \end{cases}$$

```
S - N O W Y
S U N N - Y
✓ ✗ ✓ ✗ ✗ ✓
```

Cost to align these strings: 3

```
- N O W Y
U N N - Y
✗ ✓ ✗ ✗ ✓
```

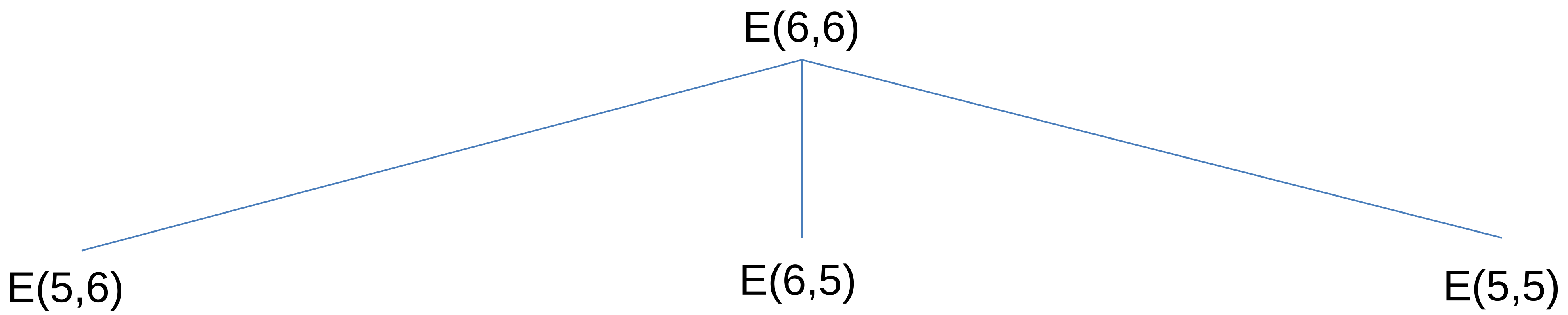
then there is not a more optimal way to align NOWY and UNNY in less than 3 edits (otherwise we would have a more optimal way to align SNOWY and SUNNY

Optimal Substructure

Edit Distance Problem

Finding **E(n,m)** requires finding all other **E**'s

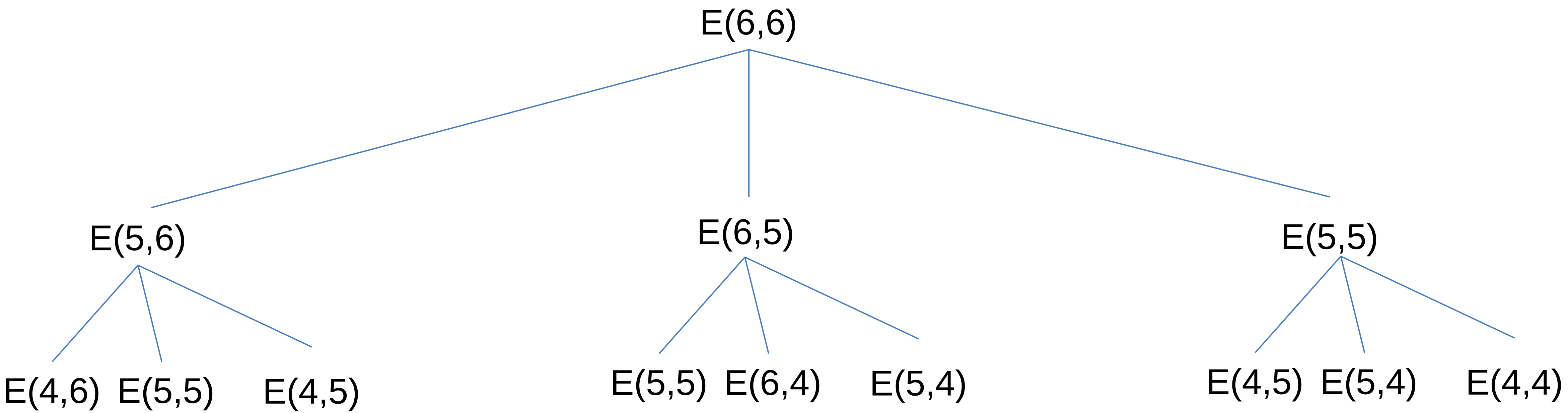
$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$



Edit Distance Problem

Finding $\mathbf{E(n,m)}$ requires finding all other $\mathbf{E}$'s

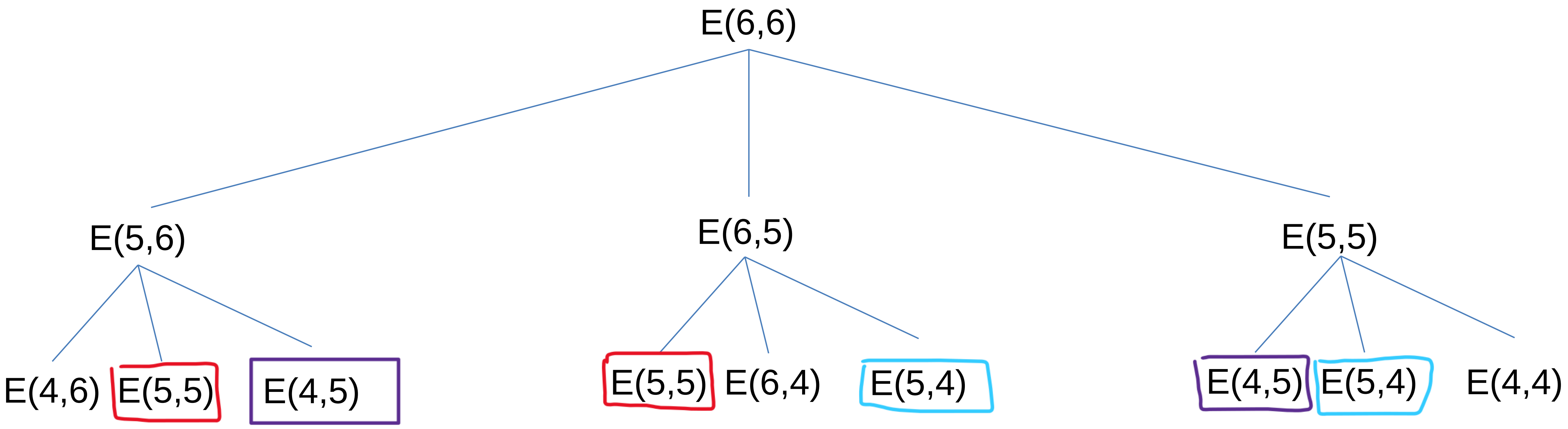
$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$



Edit Distance Problem

Finding $\mathbf{E(n,m)}$ requires finding all other $\mathbf{E}$'s

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

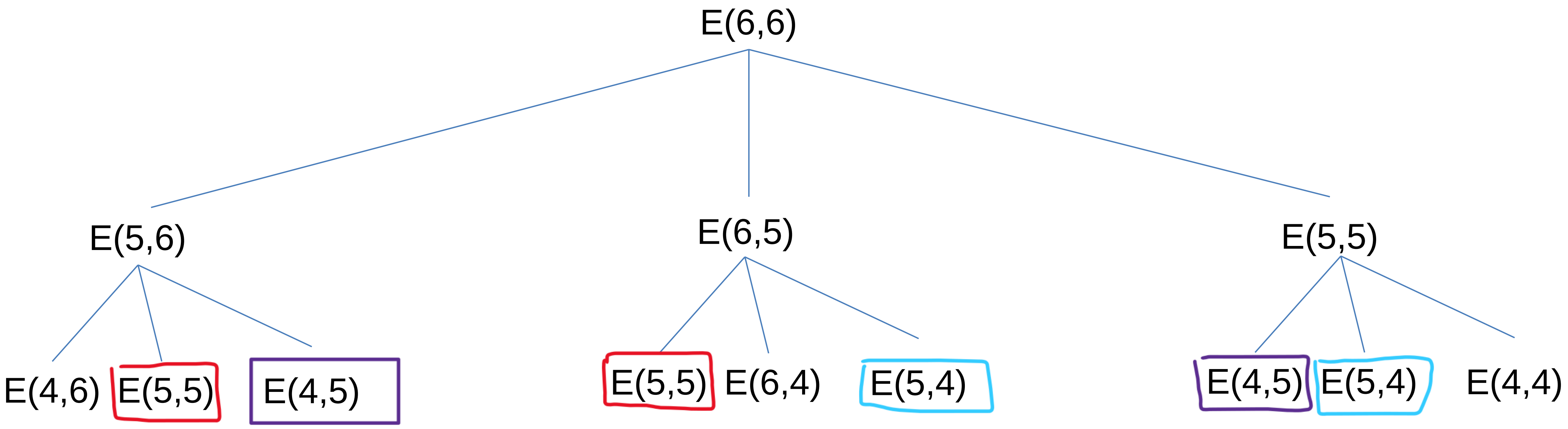


We compute the same subproblem in different branches → **overlapping subproblems**

Edit Distance Problem

Finding $\mathbf{E(n,m)}$ requires finding all other $\mathbf{E}$'s

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$



We compute the same subproblem in different branches → **overlapping subproblems**

Edit Distance Problem

x_i	$-$	x_i
$-$	y_j	y_j

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{where diff}(i,j) = \begin{cases} 0, & x_i = y_j \\ 1, & x_i \neq y_j \end{cases}$$

Finding $E(n,m)$ requires finding all the other E's, which can be represented in a 2d table with the strings along the axes.

Edit Distance

i \ j		0	1	2	3	4	5
			S	U	N	N	Y
0		0					
1	S						
2	N						
3	O					●	
4	W						
5	Y						

$E(3, 4)$

$$E(i, j) = \min \begin{cases} E(i-1, j) + 1 \\ E(i, j-1) + 1 \\ E(i-1, j-1) + \text{diff}(i, j) \end{cases}$$

$$\text{diff}(i, j) = \begin{cases} 0, & x[i] = y[j] \\ 1, & x[i] \neq y[j] \end{cases}$$

Where can we start?

Edit Distance

i \ j		0	1	2	3	4	5
			S	U	N	N	Y
0		0					
1	S						
2	N						
3	O					●	
4	W						
5	Y						

$E(3, 4)$

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i] = y[j] \\ 1, & x[i] \neq y[j] \end{cases}$$

Where can we start?

$E(0,1)$ or $E(1,0)$

Edit Distance

i		j: 0	1 (S)	2 (U)	3 (N)	4 (N)	5 (Y)
0		0					
1	S						
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(0,1) = \min \begin{cases} E(-1,1)+1 \\ E(0,0)+1 \\ E(-1,0)+1 \end{cases} = ?$$

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0					
1	S						
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i] = y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(0,1) = \min \begin{cases} \sout{E(-1,1)+1} \\ E(0,0)+1 \\ \sout{E(-1,0)+1} \end{cases} = ?$$

Edit Distance

i \ j		0	1 S	2 U	3 N	4 N	5 Y
0		0	1				
1	S						
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(0,1) = \min \begin{cases} \cancel{E(-1,1)+1} \\ E(0,0)+1 \\ \cancel{E(-1,0)+1} \end{cases} = 1$$

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1				
1	S						
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(1,1) = \min \begin{cases} E(0,1)+1 \\ E(1,0)+1 =? \\ E(0,0)+0 \end{cases}$$

Edit Distance

i \ j		0	1	2	3	4	5
			S	U	N	N	Y
0		0	1				
1	S						
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(1,1) = \min \begin{cases} E(0,1)+1 \\ \color{red}{\boldsymbol{E(1,0)}} + 1 = ? \\ E(0,0)+0 \end{cases}$$

Not calculated yet!

Edit Distance

i	j	0	1	2	3	4	5
			S	U	N	N	Y
0		0	1				
1	S						
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

Need upper left hand corner filled out before we can progress.

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2			
1	S						
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(0,2) = \min \begin{cases} E(-1,2)+1 \\ E(0,1)+1 \\ E(-1,1)+1 \end{cases} = 2$$

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2			
1	S	1					
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i] = y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(1,0) = \min \begin{cases} E(0,0)+1 \\ E(1,-1)+1 = 1 \\ E(0,-1)+1 \end{cases}$$

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2			
1	S	1	0				
2	N						
3	O						
4	W						
5	Y						

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

$$E(1,1) = \min \begin{cases} E(0,1)+1 \\ E(1,0)+1 = 0 \\ E(0,0)+0 \end{cases}$$

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

Running Time?

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

$$\text{diff}(i,j) = \begin{cases} 0, & x[i]=y[j] \\ 1, & \text{otherwise} \end{cases}$$

Running Time?

Fill out $n \times m$ table with constant operations: $O(nm)$

Edit Distance

		j	0	1	2	3	4	5
i				S	U	N	N	Y
0			0	1	2	3	4	5
1	S		1	0	1	2	3	4
2	N		2	1	1	1	2	3
3	O		3	2	2	2	2	3
4	W		4	3	3	3	3	3
5	Y		5	4	4	4	4	**3**

Edit distance = **3**.

How can we recreate the actual alignments?

Backtracking.

Ask the question: “How did we get here?”

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	**3**

How did we get to $E(5,5)$?

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	← **3**

How did we get to $E(5,5)$?

From $E(5,4)$?

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	← **3**

How did we get to $E(5,5)$?

From $E(5,4)$? – No. Can never go down in cost.

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	**3** ↑

How did we get to $E(5,5)$?

From $E(5,4)$? – No. Can never go down in cost.

From $E(4,5)$?

Edit Distance

i \ j		0	1 S	2 U	3 N	4 N	5 Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3 ↑

How did we get to $E(5,5)$?

From $E(5,4)$? – No. Can never go down in cost.

From $E(4,5)$? – No. Need +1 to move that direction.

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

Edit Distance

i \ j		0	1	2	3	4	5
			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	**3**

How did we get to $E(5,5)$?

- From $E(5,4)$? – No. Can never go down in cost.
- From $E(4,5)$? – No. Need +1 to move that direction.
- From $E(4,4)$?

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

Edit Distance

j		0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	**3**

How did we get to $E(5,5)$?

From $E(5,4)$? – No. Can never go down in cost.

From $E(4,5)$? – No. Need +1 to move that direction.

From $E(4,4)$? – Yes. Match Y's.

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Continuing the process yields all of the optimal solutions.

Diagonal move indicates ?

Vertical move indicates ?

Horizontal move indicates ?

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\mathrm{diff}(i,j) \end{cases}$$

Edit Distance

Continuing the process yields all of the optimal solutions.

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Diagonal move indicates match.

Vertical move indicates ?

Horizontal move indicates ?

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Continuing the process yields all of the optimal solutions.

Diagonal move indicates match.

Vertical move indicates space inserted in j.

Horizontal move indicates ?

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Continuing the process yields all of the optimal solutions.

Diagonal move indicates match.

Vertical move indicates space inserted in j.

Horizontal move indicates space inserted in i.

$$E(i,j) = \min \begin{cases} E(i-1,j)+1 \\ E(i,j-1)+1 \\ E(i-1,j-1)+\text{diff}(i,j) \end{cases}$$

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Diagonal move indicates match.

Vertical move indicates space inserted in j.

Horizontal move indicates space inserted in i.

S – N O W Y

S U N N – Y

Edit Distance

i	j	0	1	2	3	4	5
			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Diagonal move indicates match.

Vertical move indicates space inserted in j.

Horizontal move indicates space inserted in i.

Alignment?

Edit Distance

i \ j		0	1	2	3	4	5
			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Diagonal move indicates match.

Vertical move indicates space inserted in j.

Horizontal move indicates space inserted in i.

S – N O W Y

S U N – N Y

Edit Distance

	j	0	1	2	3	4	5
i			S	U	N	N	Y
0		0	1	2	3	4	5
1	S	1	0	1	2	3	4
2	N	2	1	1	1	2	3
3	O	3	2	2	2	2	3
4	W	4	3	3	3	3	3
5	Y	5	4	4	4	4	3

Diagonal move indicates match.

Vertical move indicates space inserted in j.

Horizontal move indicates space inserted in i.

S N O W Y

S U N N Y

Applications of Edit Distance

GGAACAGATTGGTCTAATTAGCTTAAGAGAGTAAATTCTGGGATCATTCA

GTAGTAATCACAAATTTACGGTGGGGCTTTTTTGGCGGATCTTTACAGAT

Edit Distance (Levenshtein Distance): 29

Edit Distance is used in computational biology to find how similar two sequences of DNA are

Applications of Edit Distance

We can could use edit distance to correct misspelled words!

`mawntain`

Did you mean **`maintain`** (1) ?
Did you mean **`mountain`** (2) ?
Did you mean **`captain`** (3) ?
Did you mean **`mantis`** (3) ?