5 February 2025

Dynamic Programming III
Analysis of Edit Dist

Plan

* Review Edit Distance Problem
* Announcements
* Proofs of Alignment
* Extensions

Edit Distance

Given two strings S, T

Find minimum sequence of edits

to turn S into T

S = CRYPTOGRAPHY

T = ENCRYPTING

Edit Distance

Given two strings S, T

Find minimum sequence of edits

to turn S into T

S = C R Y P T O G R A P H Y

T = E N C R Y P T I N G

Addition to T (cost γ)

e.g. $\gamma = 1$

Edit Distance

Given two strings S, T

Find minimum sequence of edits to turn S into T

Deletion from S

S = C R Y P T O G R A P ~~H~~ ~~Y~~

T = E N C R Y P T I N G

(cost γ)

e.g. $\gamma = 1$

Edit Distance

Given two strings S, T

Find minimum sequence of edits to turn S into T

$S =$ C R Y P T O G R A P H Y

$T =$ E N C R Y P T I N G

Change $S_i \longrightarrow T_j$

Cost $\Delta_{S_i T_j}$

Edit Distance

Given two strings S, T

Find minimum sequence of edits

to turn S into T

S = C R Y P T O G R A P H Y

T = E N C R Y P T I N G

Edit Distance

Given two strings S, T

Find minimum sequence of edits

to turn S into T

Announcements

* HW1 Due
* HW2 Out Today
* Prelim #1
  - Next Thurs. 13 Feb 7:30-9p.
  - Closed Note, Closed Internet.
* Review Session
  - Saturday Recitation
  - Next Tues. 11 Feb 7-9p
    Gates G01

Friday

* Guest Lecture by Prof. Bobby Kleinberg

* Prof. Kim's OH moved to Monday.

Edit Distance

Given two strings S, T

Find minimum sequence of edits to turn S into T

S = C R Y P T O G R A P H Y

T = E N C R Y P T I N G

What do we know about the optimal solution?

Edit Distance

Given two strings S, T

Find minimum sequence of edits to turn S into T

S = C R Y P T O G R A P H Y

....

T = E N C R Y P T I N G

Fact 1. In every sequence of edits, each character in S is either

* Deleted
* Changed/Aligned into T

Edit Distance

Given two strings S, T

Find minimum sequence of edits to turn S into T

S = C R Y P T O G R A P H Y

T = E N C R Y P T I N G

Fact 2. In every sequence of edits, each character in T is either

* Added
* Changed/Aligned from S

Fact 1 + Fact 2 $\implies$ Exhaustive Case Analysis

Cases

* S_n is deleted
* T_m is added
* At least one of them is aligned/changed.

Fact 1 + Fact 2 ⟹ Exhaustive Case Analysis

S_n deleted

C R Y P T O ~~G~~

E N C R Y

edit distance between S and T $\leq$ edit distance between S[1:n-1] and T $+ \gamma$

Fact 1 + Fact 2 ⟹ Exhaustive Case Analysis

T_m is added

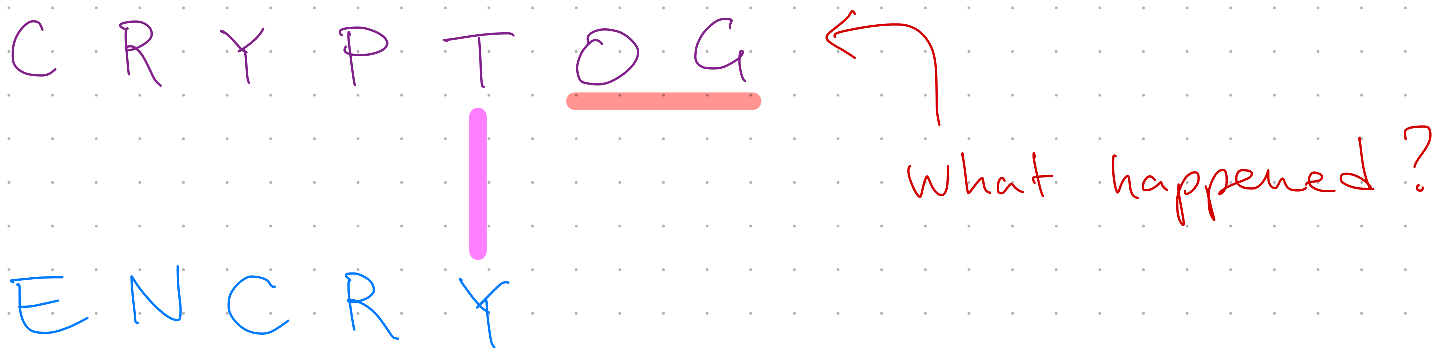
C R Y P T O G

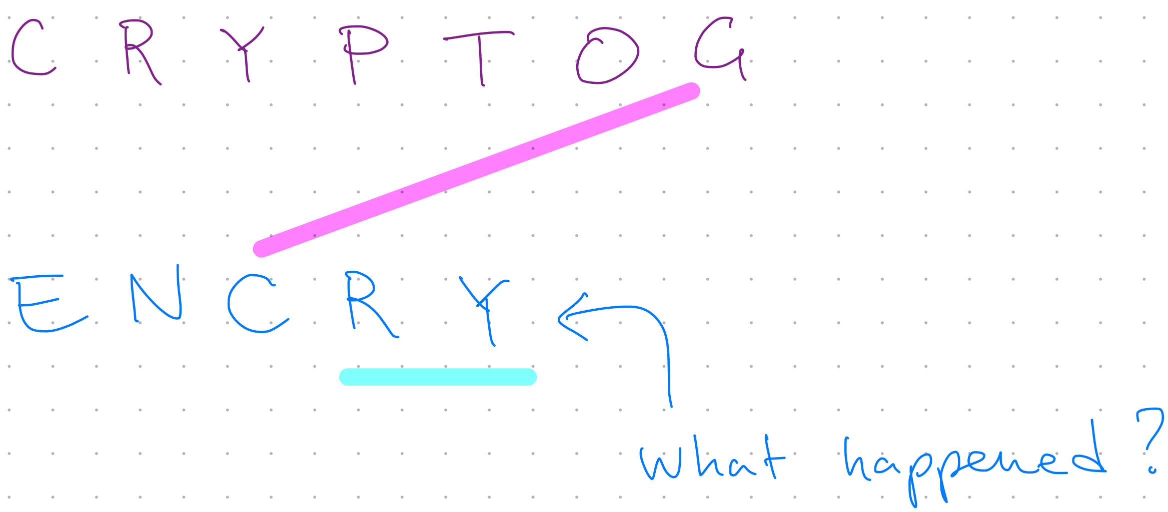
E N C R Y

edit distance between S and T $\leq$ edit distance between S and T[1:m-1] $+ \gamma$

Fact 1 + Fact 2 ⟹ Exhaustive Case Analysis

At least one aligned ?





Fact 1 + Fact 2 ⟹ Exhaustive Case Analysis

At least ~~one~~ aligned?

S_n aligned w/ T_m

C R Y P T O G

E N C R Y

edit distance between S and T $\leq$ edit distance between S[1:n-1] and T[1:m-1] $+ \Delta_{S_n T_m}$

Fact 1 + Fact 2 $\Rightarrow$ Exhaustive Case Analysis

Cases

* S_n is deleted
* T_m is added
* S_n aligned to T_m

↓

DP Recurrence

Edit Distance between S and T = min
- γ + Edit distance between $S[1:n-1]$ and T
- γ + Edit distance between S and $T[1:m-1]$
- $\Delta_{S_n T_m}$ + Edit distance between $S[1:n-1]$ and $T[1:m-1]$

Correctness. Consider the recurrence ED given as

$$ED(n,m) = \min \begin{cases} ED(n-1, m-1) + \Delta_{S_n T_m}, \\ ED(n-1, m) + \gamma, \\ ED(n, m-1) + \gamma \end{cases}$$

where

$$ED(0,0) = 0, \quad ED(1,0) = \gamma, \quad ED(0,1) = \gamma.$$

Theorem.

For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

$$\boxed{\text{Edit Distance between } S \text{ and } T} = ED(n, m).$$

Edit Distance Algorithm

$ED(0,0) = 0$

$ED(i,0) = i \cdot \gamma$

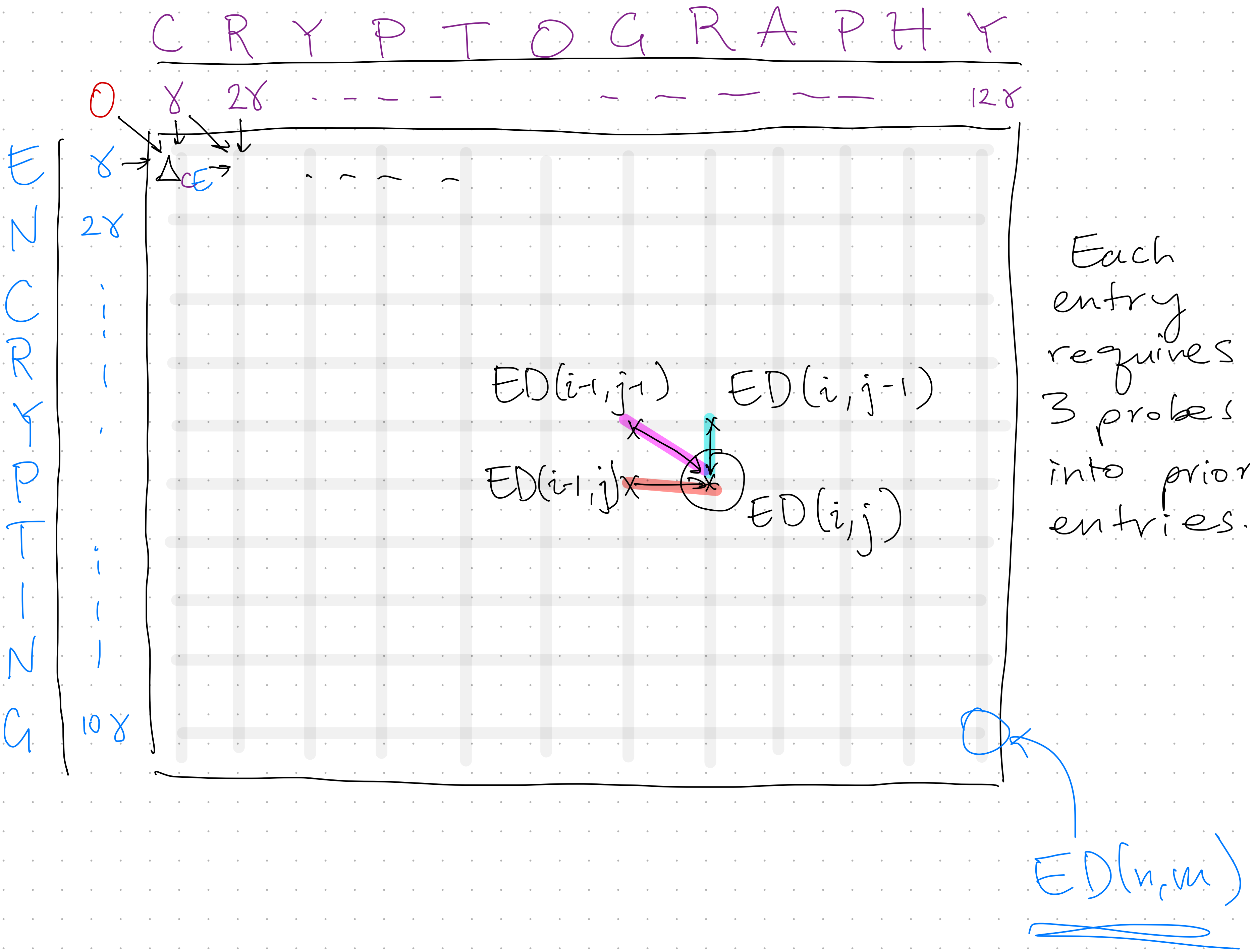
$ED(0,j) = j \cdot \gamma$

For $i = 1 \rightarrow n$

For $j = 1 \rightarrow m$

$$ED(i,j) = \min \begin{cases} ED(i-1, j-1) + \Delta_{S_i T_j}, \\ ED(i-1, j) + \gamma, \\ ED(i, j-1) + \gamma \end{cases}$$

return $ED(n,m)$



C R Y P T O G R A P H Y
0
γ
2γ
12γ
E N C R Y P T I N G
γ
2γ
10γ
Δ CE
ED(i-1, j-1)
ED(i, j-1)
ED(i-1, j)
ED(i, j)
Each entry requires 3 probes into prior entries.
ED(n, m)

Edit Distance Algorithm

$ED(0,0) = 0$
$ED(i,0) = i \cdot \gamma$
$ED(0,j) = j \cdot \gamma$

Theorem. The Edit Distance DP Algorithm runs in $O(nm)$ time

For $i = 1 \to n$
    For $j = 1 \to m$
        $$ED(i,j) = \min \begin{cases} ED(i-1, j-1) + \Delta_{S_i T_j}, \\ ED(i-1, j) + \gamma, \\ ED(i, j-1) + \gamma \end{cases}$$

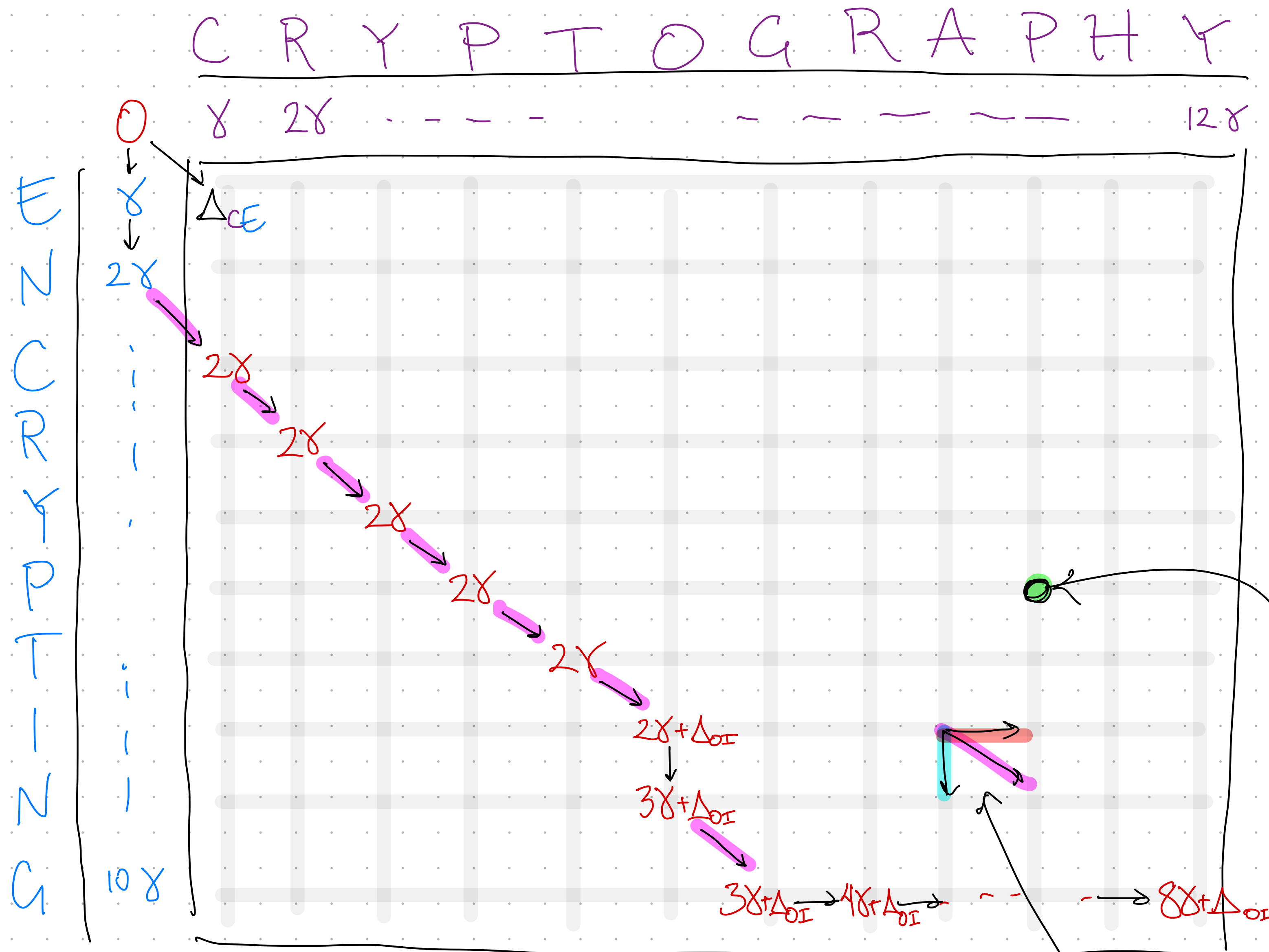
return $ED(n,m)$

Edit Distance ≡ Sequence Alignment

An *alignment* is a non-crossing matching.

ED Algorithm computed the *distance*

Can we compute the **alignment**?



Idea.

Treat DP Table as a graph

entries = vertices

transitions = edges

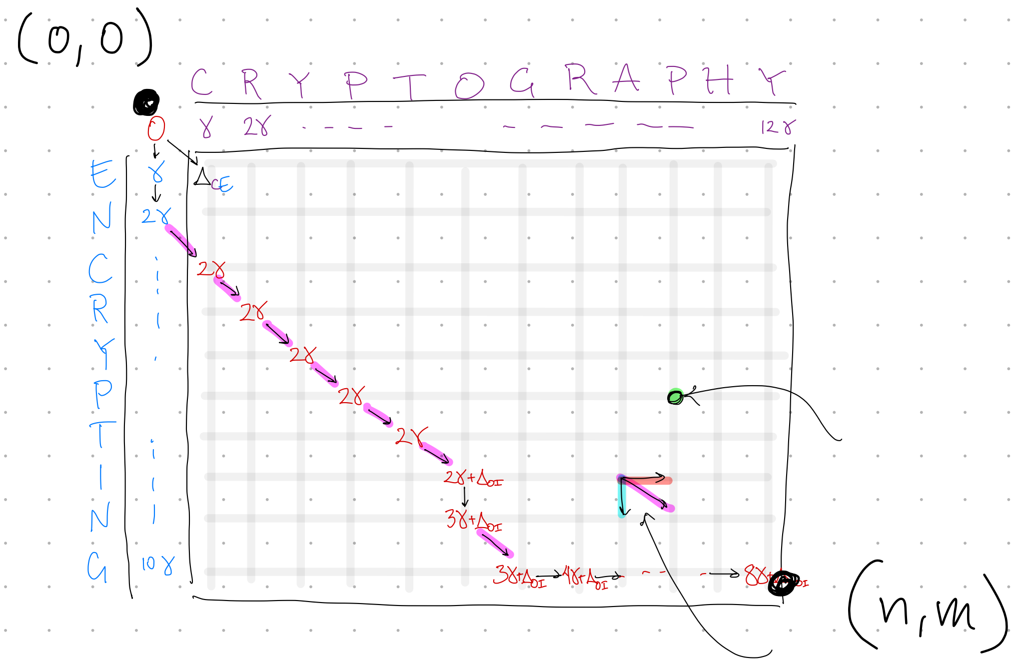
min ED alignment = shortest path

diagonal edges = alignment.

Theorem:

For any $n, m \in \mathbb{N}$. Suppose $|S| = n$, $|T| = m$.

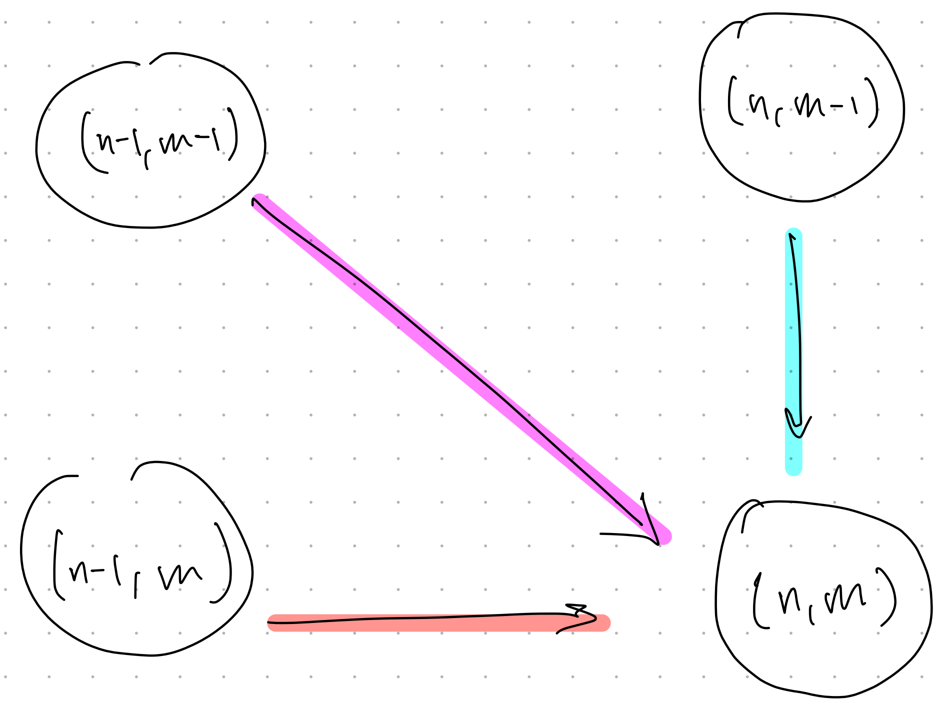
Edit Distance between S and T = **shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph**



$\Rightarrow$ Can reconstruct alignment via diagonal edges.

Fact 3 Every path from $(0,0)$ to (n,m) in the ED Graph uses one of three edges:

- $(n-1, m) \longrightarrow (n,m)$
- $(n, m-1) \longrightarrow (n,m)$
- $(n-1, m-1) \longrightarrow (n,m)$



Theorem:

For any $n, m \in \mathbb{N}$. Suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Proof By induction on $n+m$.

* Two base cases (direct analysis)

* Inductive Step (based on ED Recurrence & Fact 3)

For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T	=	Shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Base Cases $n + m = 0$, and $n + m = 1$

When $n + m = 0$, $|S| = |T| = 0$,

so both strings equal the empty string.

$S = $ "" $\quad T = $ ""

For any $n, m \in \mathbb{N}$. Suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Base Cases $n+m=0$, and $n+m=1$

When $n+m=0$, $|S| = |T| = 0$,

so both strings equal the empty string.

$S = ""$ $T = ""$

Edit Distance between "" and "" $= 0 =$ shortest path $(0,0) \rightsquigarrow (0,0)$ in ED Graph

For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T	=	Shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Base Cases $\quad n+m = 0$, and $n+m = 1$

When $n+m=1$, then $|S|=1$ and $|T|=0$.
(or vice versa)

S = "~~C~~" $\quad T$ = ""

S can be turned to T via one deletion

Edit Distance between "C" and ""	= γ

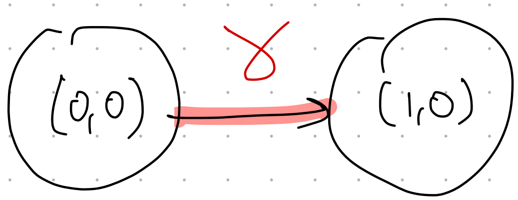
For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Base Cases $n+m = 0$, and $n+m = 1$

* The vertex $(1,0)$ has 1 in-edge
* from $(0,0)$ w/ length γ



$\Rightarrow$ shortest path $(0,0) \rightsquigarrow (1,0)$ in ED Graph $= \gamma$

For any $n, m \in \mathbb{N}$. Suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = Shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Base Cases $n+m = 0$, and $n+m = 1$

When $n+m=1$, then $|S|=1$ and $|T|=0$.

Edit Distance between "C" and "" $= \gamma =$ Shortest path $(0,0) \rightsquigarrow (1,0)$ in ED Graph

$(0,0) \rightsquigarrow (0,1)$ similar

For any $n, m \in \mathbb{N}$. Suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = shortest path $(0,0) \leadsto (n,m)$ in ED Graph

---

Inductive Step

IH. $\forall i, j$ such that $i+j \leq n+m$

Edit Distance between $S[1:i]$ and $T[1:j]$ = shortest path $(0,0) \leadsto (i,j)$ in ED Graph

For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = Shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Inductive Step

First, Recall ED Recurrence.

Edit Distance between S and T = min {
- γ + Edit distance between $S[1:n-1]$ and T
- γ + Edit distance between S and $T[1:m-1]$
- $\Delta_{S_n T_m}$ + Edit distance between $S[1:n-1]$ and $T[1:m-1]$

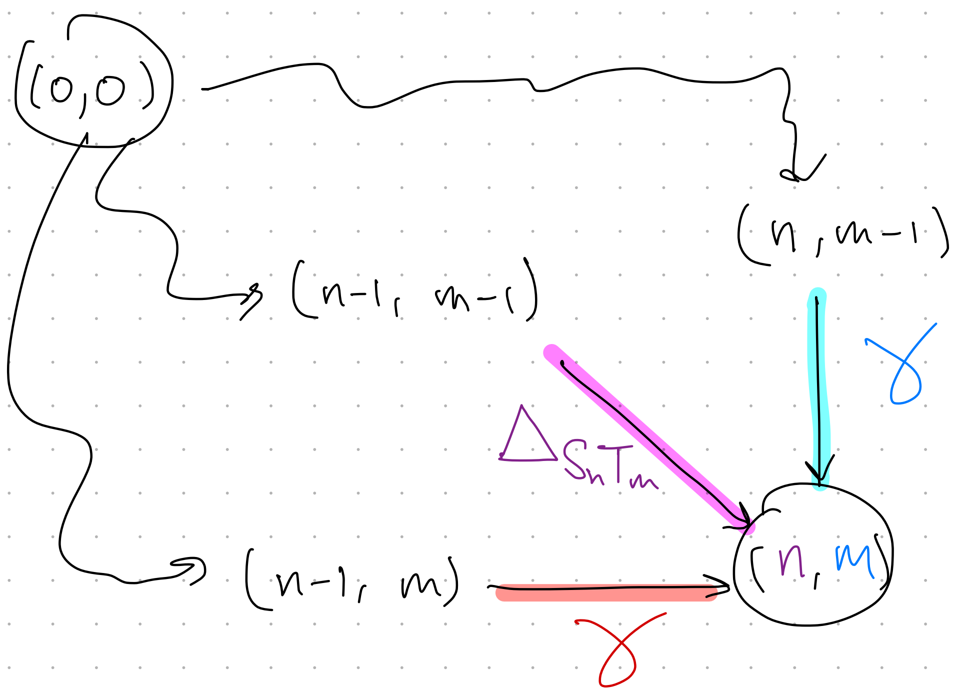
For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = Shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Inductive Step

By Fact 3, (n,m) has 3 in-edges, so shortest path is the min of in-edge wts + shortest path to neighbor.



= Shortest path from $(0,0) \rightsquigarrow (n,m)$

For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T = Shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Inductive Step

By Fact 3, (n,m) has 3 in-edges, so shortest path is the min of in-edge wts + shortest path to neighbor.

$$\min \begin{cases} \delta + \text{Shortest path from } (0,0) \rightsquigarrow (n-1, m) \\ \delta + \text{Shortest path from } (0,0) \rightsquigarrow (n, m-1) \\ \Delta_{S_n T_m} + \text{Shortest path from } (0,0) \rightsquigarrow (n-1, m-1) \end{cases} = \text{Shortest path from } (0,0) \rightsquigarrow (n, m)$$

For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

Edit Distance between S and T	=	Shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph

---

Inductive Step

By IH. Replace ED w/ SP for $i+j < n+m$

$\gamma +$	Edit distance between $S[1:n-1]$ and T	=	$\gamma +$	Shortest path from $(0,0) \rightsquigarrow (n-1, m)$
$\gamma +$	Edit distance between S and $T[1:m-1]$	=	$\gamma +$	Shortest path from $(0,0) \rightsquigarrow (n, m-1)$
$\Delta_{S_n T_m} +$	Edit distance between $S[1:n-1]$ and $T[1:m-1]$	=	$\Delta_{S_n T_m} +$	Shortest path from $(0,0) \rightsquigarrow (n-1, m-1)$

$$\Longrightarrow \quad ED(S, T) = SP\ (0,0) \rightsquigarrow (n, m)$$

∎

Theorem.

For any $n, m \in \mathbb{N}$. suppose $|S| = n$, $|T| = m$.

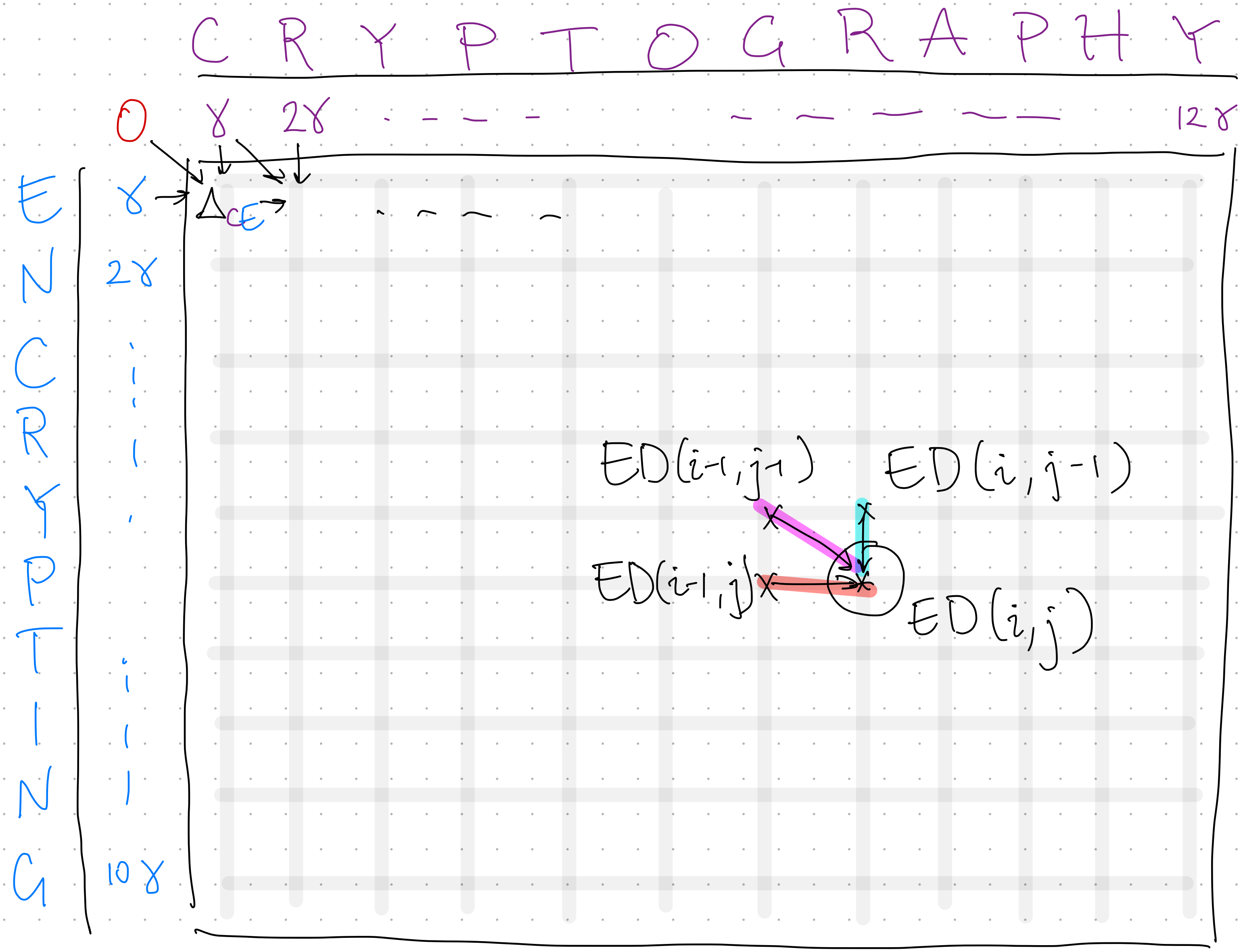
Edit Distance between S and T = shortest path $(0,0) \rightsquigarrow (n,m)$ in ED Graph ✓

In other words, Edit Distance reduces to single source shortest path.

↳ Bellman-Ford generalizes ED.

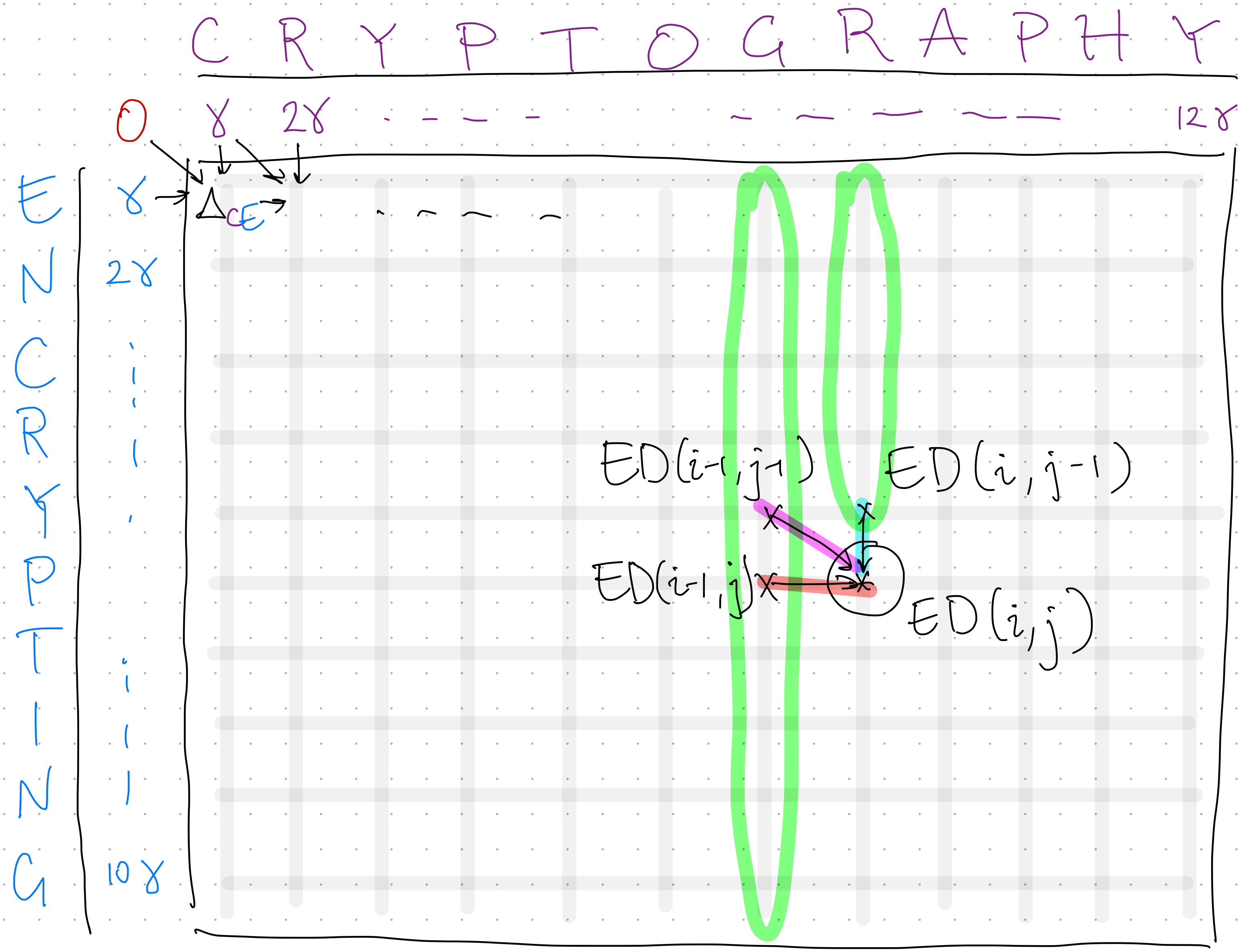
Reducing the Space Complexity?

Table $O(n \cdot m)$ entries.



Each entry requires 3 probes into prior entries.

Reducing the Space Complexity?



Each entry requires 3 probes into prior entries.

↳ From prev. column from top to bottom.

Linear Space ED.

$Prev(j) = j \cdot \gamma$ // m-entry 1D arrays

$Curr(j) = 0$

For $i = 1 \longrightarrow n$

  $Curr(0) = i \cdot \gamma$

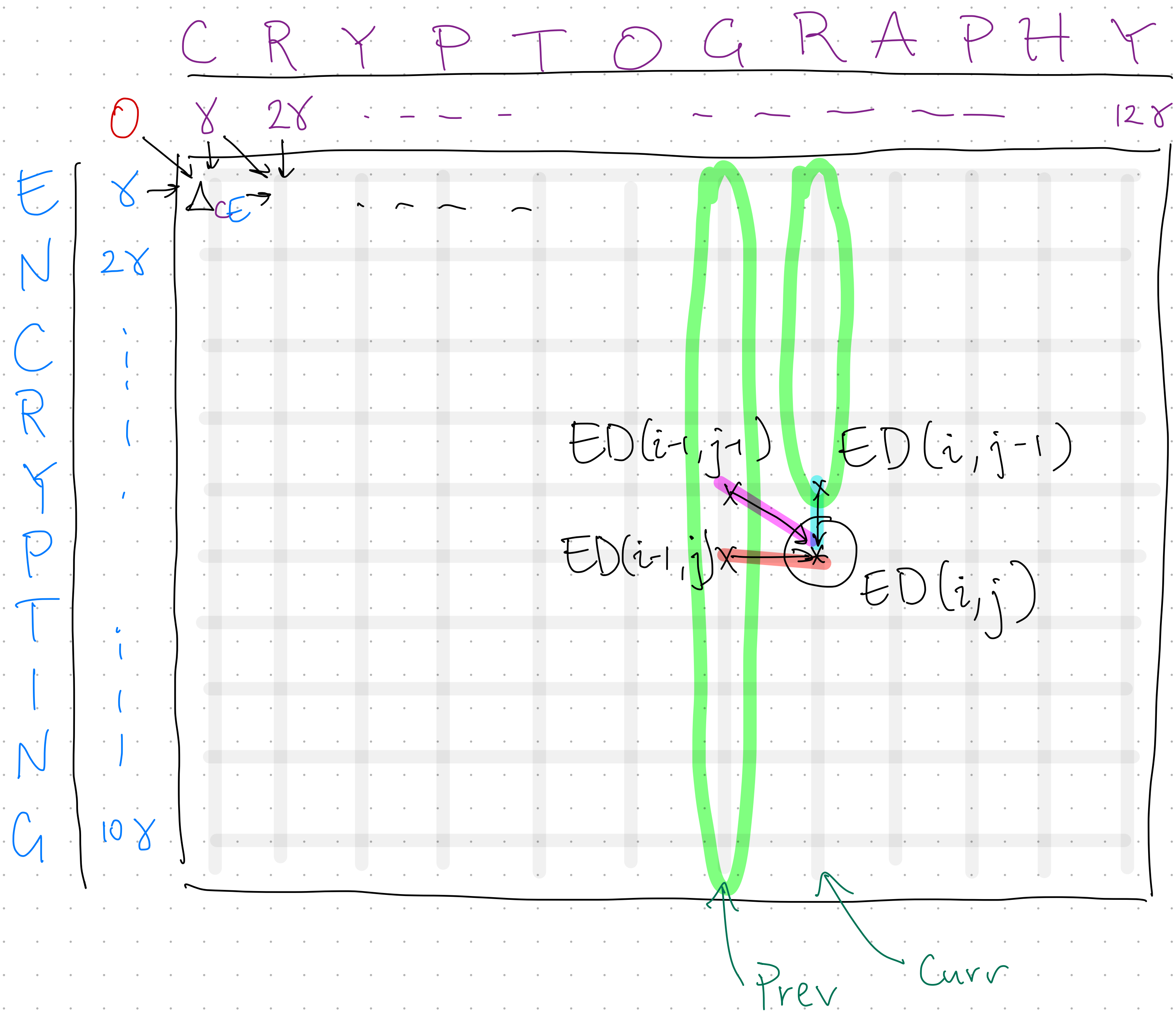
  For $j = 1 \longrightarrow m$

    $Curr(j) = \min\{ Prev(j-1) + \Delta_{S_i T_j},\ Prev(j) + \gamma,\ Curr(j-1) + \gamma \}$

  $Prev \leftarrow Curr.$

return $Curr(m)$.

Reducing the Space Complexity?



Each entry requires 3 probes into prior entries.

↳ From prev. column from top to bottom.

Theorem There exists an algorithm solving the Edit Distance problem running in $O(n \cdot m)$ time using $O(\min\{n, m\})$ space.