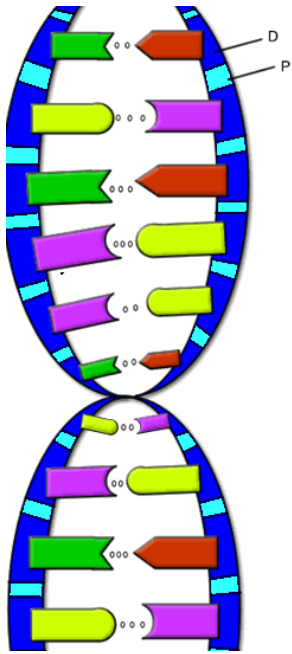


Nucleotide Sequence Alignment Part I

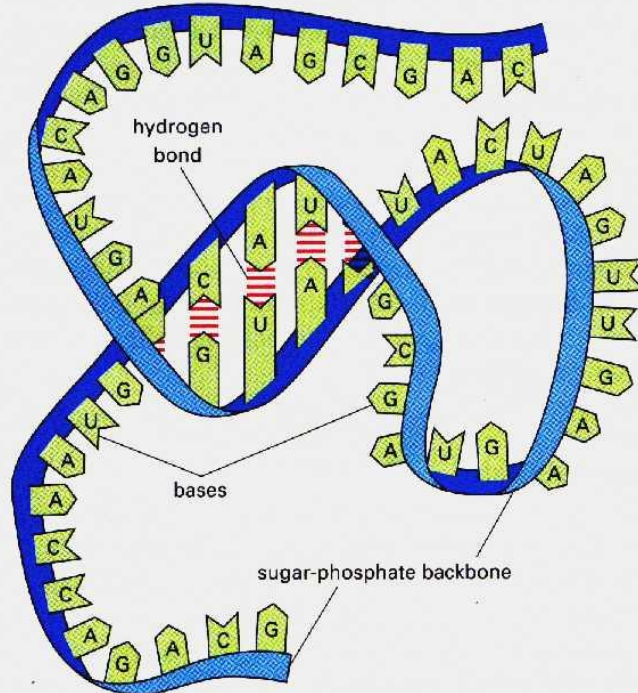
Some slides contributed by Prof. Daniel Lopresti

The Big Picture

- Three of the most important kinds biological molecules are fundamentally sequential



DNA



RNA



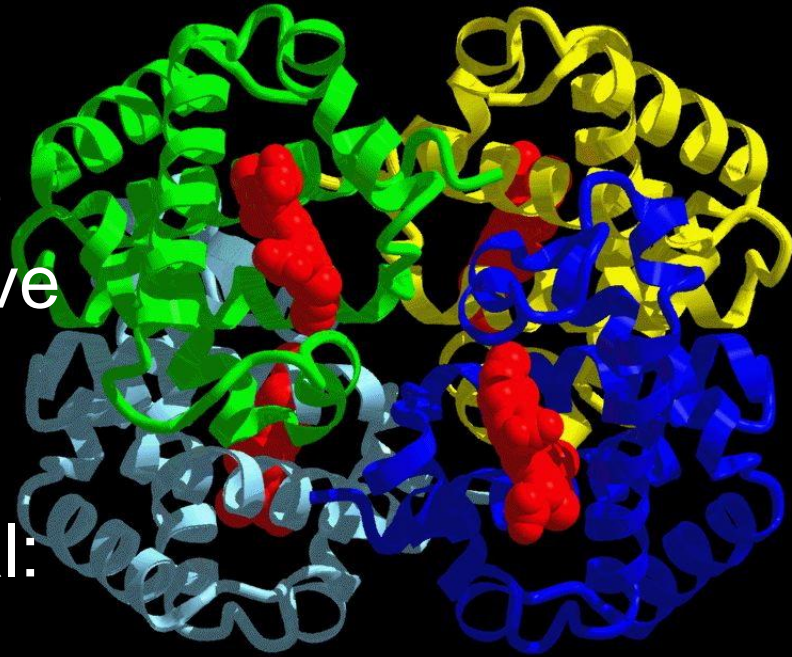
Protein

Subtle differences in sequence are crucial

- Single Nucleotide Polymorphisms
 - Individual changes in DNA sequences can impart huge medical problems
 - Sickle Cell Disease
 - Thalassemia (another hemoglobin misfolding disease)
- Antibiotic Resistance Mutations
 - Individual amino acids in bacteria mutate to cell-wall forming proteins from penicillins
- Viral Resistance Mutations
 - Individual amino acids in HIV change to destabilize drug binding, like AZT, a classic protease inhibitor.

Sequence is more than just medicine

- Evolution is the competitive selection of random changes in evolutionary populations
- The activity of selective pressure is visible in the way sequences change in evolution:
 - Mutations happen anywhere
 - Organisms with uncompetitive mutations don't reproduce
 - Mutations in oxygen binding in Hemoglobin are often fatal: organism can't transport oxygen
- Mutations elsewhere might not matter



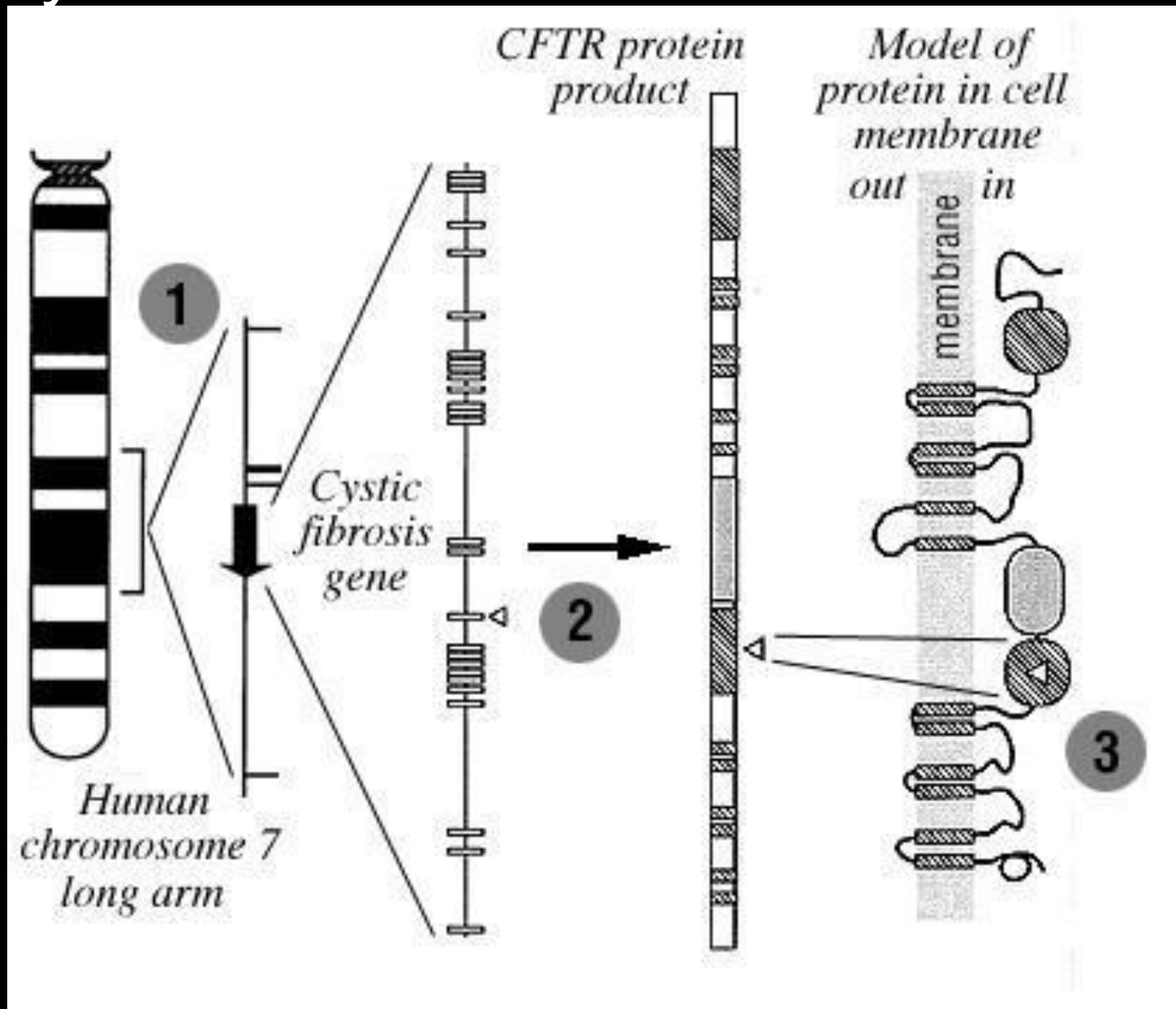
Sequence similarities can reveal function

- When we look at DNA and protein sequences, we don't know what those molecules do in a biological system – but we want to know!
- If we find genes or proteins that are similar to genes or proteins we've seen before, then this tells us what the function could be.
- In the applications project, you will find exact matches, but if you could only find approximate similarity, those are hints about gene function

The story of cystic fibrosis

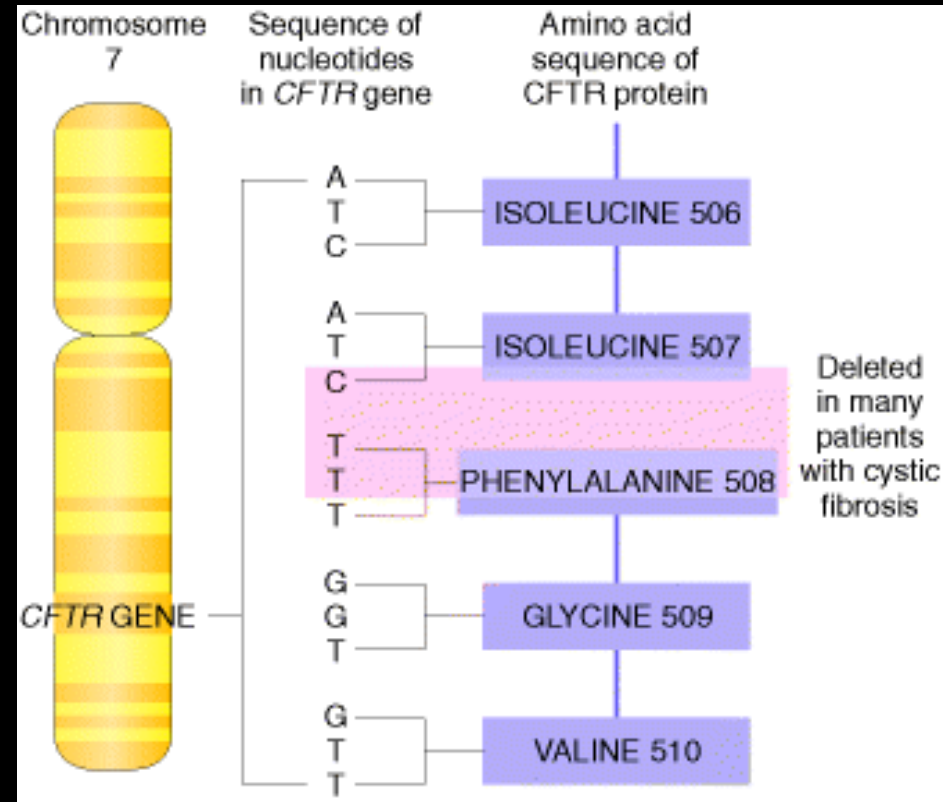
- Hereditary disorder that affects the chemical components in sweat, mucus and several other bodily fluids
- Incorrect viscosity in essential mucus leads to many macroscopic challenges:
 - Maladsorption of food and vitamins
 - Mucus accumulation in the lungs, leading too many cardiovascular challenges
- Observed as autosomal recessive in the 1980s
 - One chromosome without CF mutations avoids symptoms.

The Cystic Fibrosis Gene



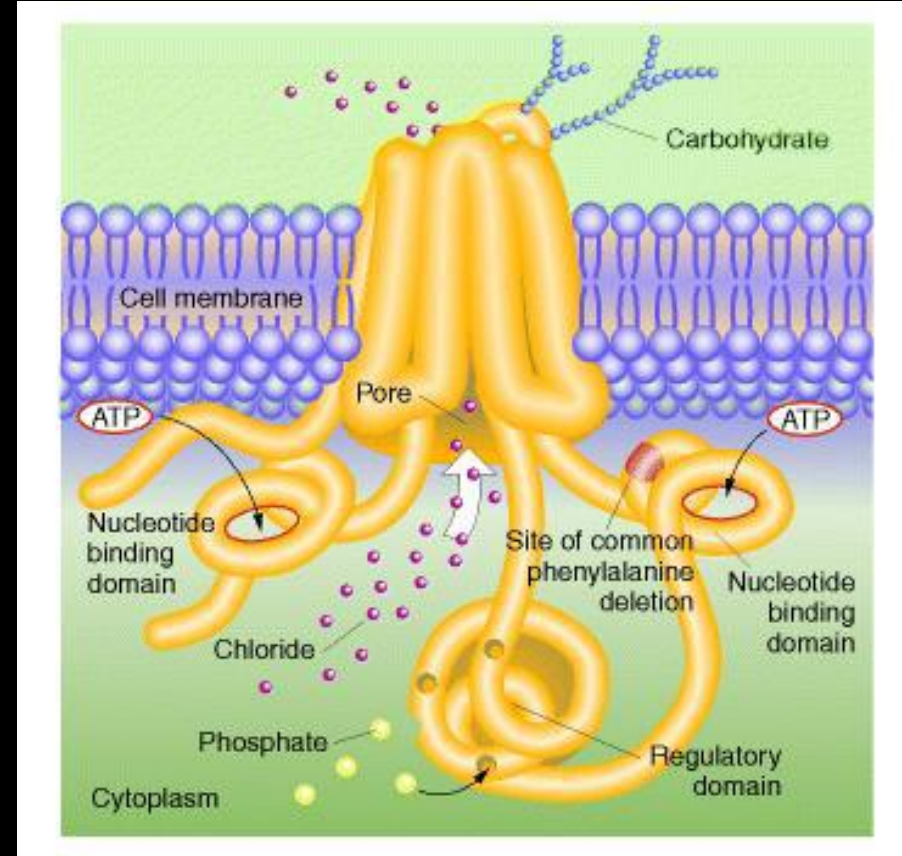
Mutations in the Cystic Fibrosis Gene

- 70% of CF patients exhibit a deletion that affects one amino acids.
- Deletes phenylalanine residue in the Cystic Fibrosis Transmembrane Conductance Regulator (CFTR)



A picture of CFTR

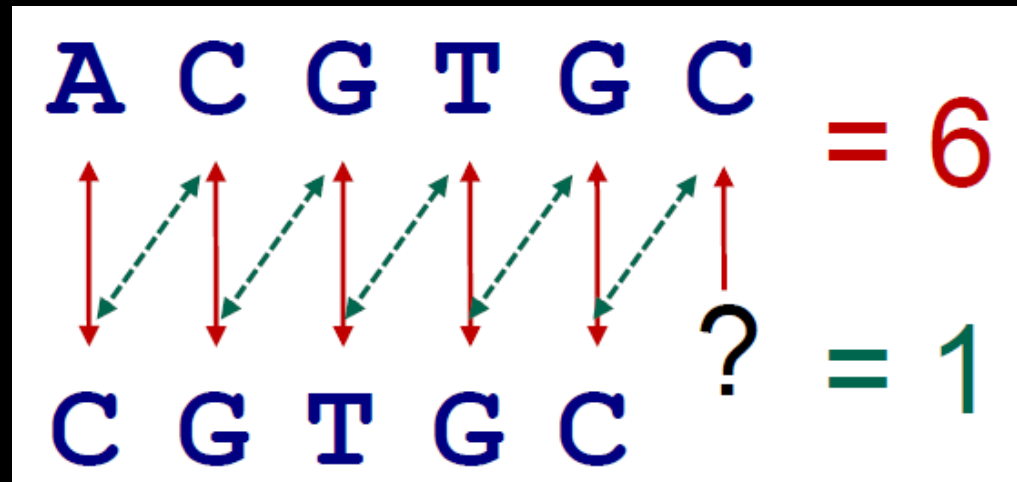
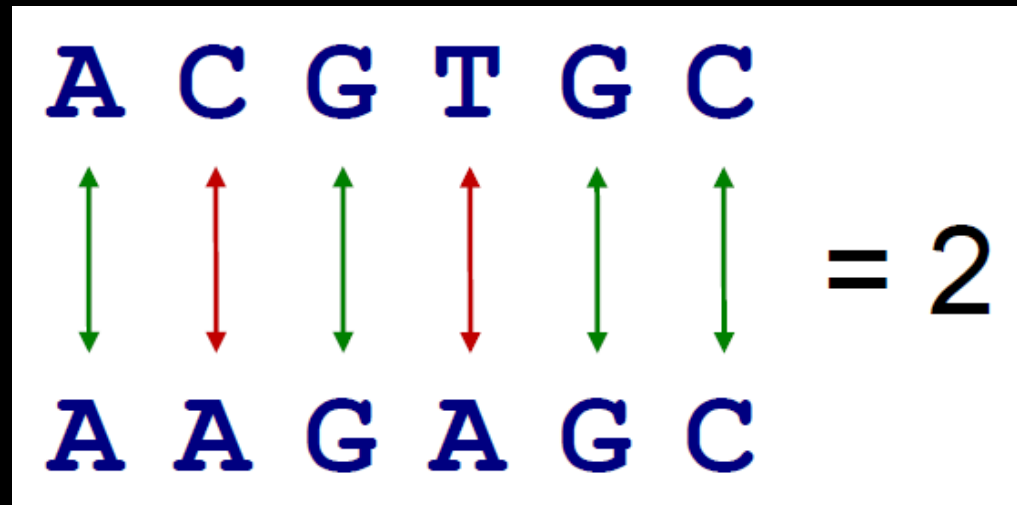
- Expressed in epithelial cells that secrete mucus
- Line airways, digestive tract, etc.
- Adjusts the amount of water in cell secretions
- In CF, CFTR insufficient water is released, leading to mucus that is too thick



We want to find mutations like this automatically

- DNA, RNA, and protein sequences are long for humans and sometimes repetitive
- If we cannot automatically compare sequences, then sequencing is pointless
- As our sequencing technology improves, we need to be able to compare more sequences
- Fortunately, we have good ways to compare sequences

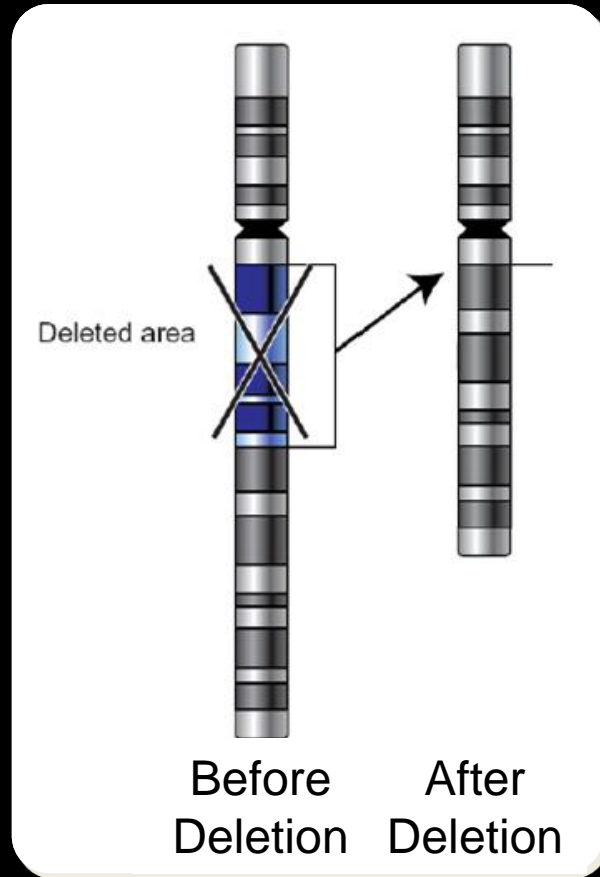
A bad way to do sequence comparison



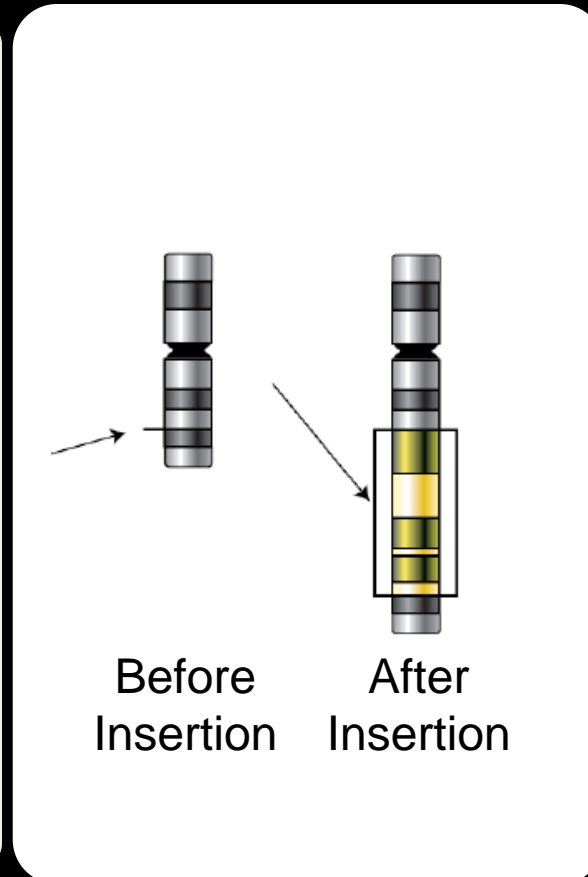
- We can look at two sequences, and count the number of identical pairs, and subtract the different pairs
- Problem: a tiny shift lead can make a big difference

Sequence similarity for biological purposes

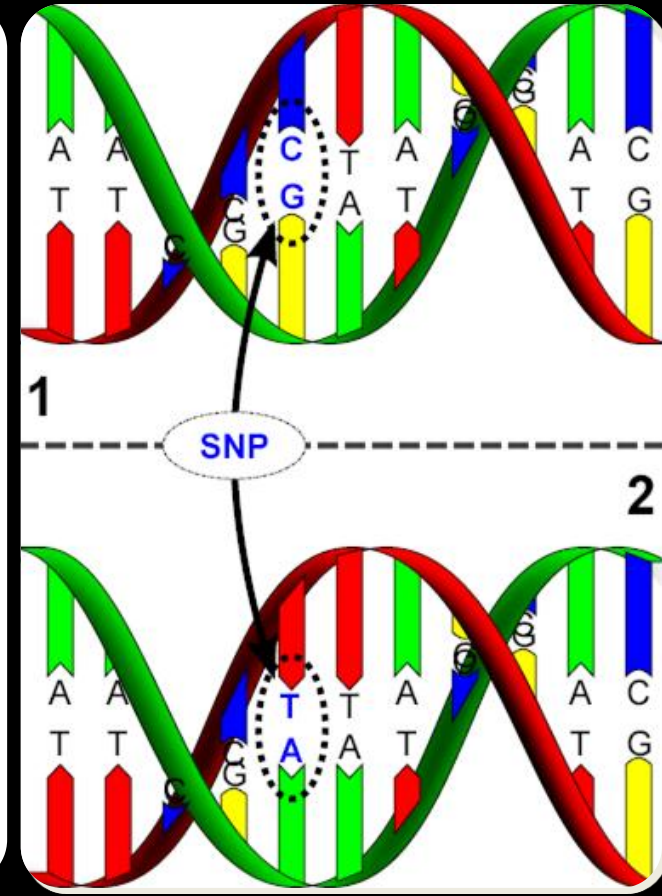
- We need to measure similarity based on changes that happen in biological systems.



Deletion



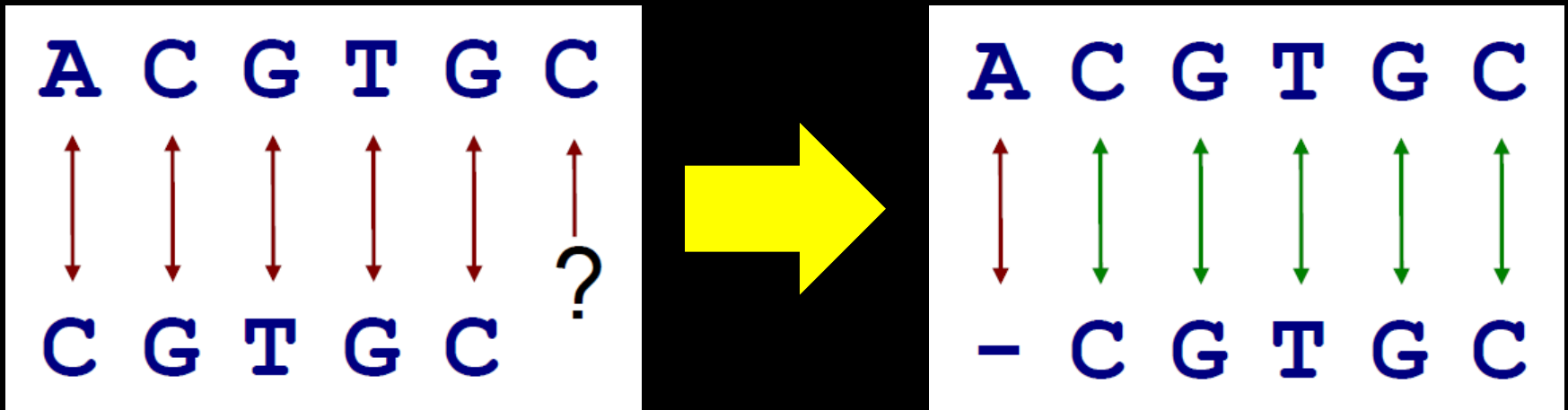
Insertion



SNPs

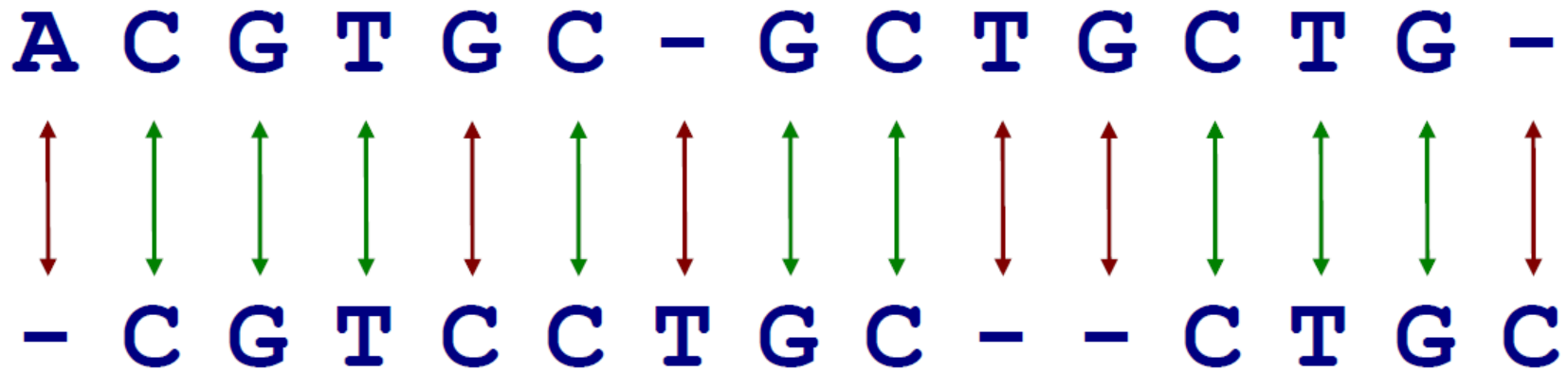
What is a sequence alignment

- An alignment is a monotonic bijection between two sequences mapping every member of one sequence to a member of the other sequence, or to a “gap”.



- Monotonic: Arrows cannot cross. If A before B in sequence 1, A before B in the alignment

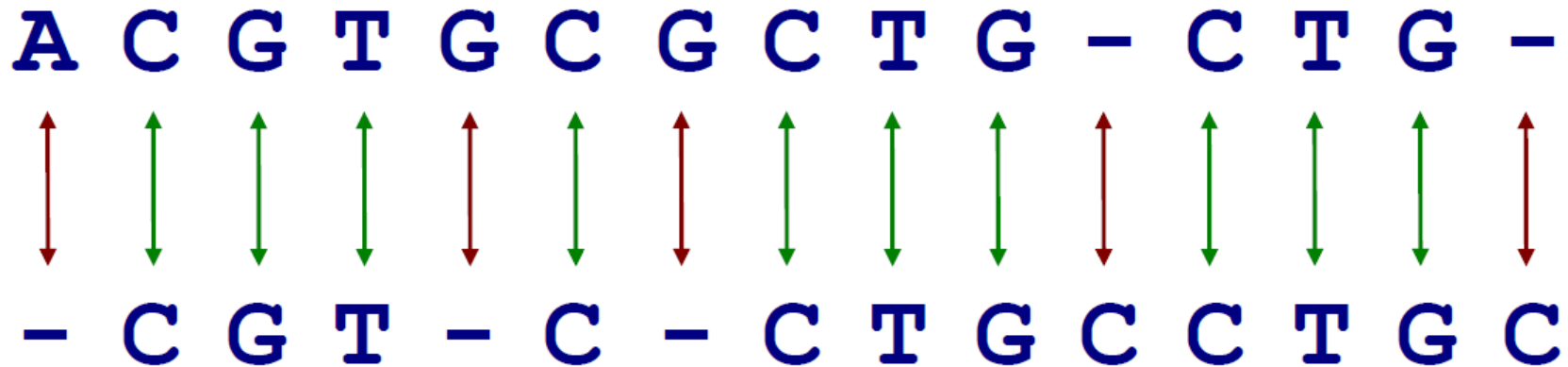
Optimal Sequence Alignments



A sequence alignment diagram showing two DNA sequences. The top sequence is A C G T G C - G C T G C T G -. The bottom sequence is - C G T C C T G C - - C T G C. Vertical arrows connect the characters: red arrows indicate mismatches (A to -, G to C, G to C, G to -, C to C, C to T, T to G, T to -, G to C) and green arrows indicate matches (C to C, T to T, C to C, C to T, G to G, C to C, T to T, G to G).

A C G T G C - G C T G C T G -
- C G T C C T G C - - C T G C

- Does this alignment maximize identity?



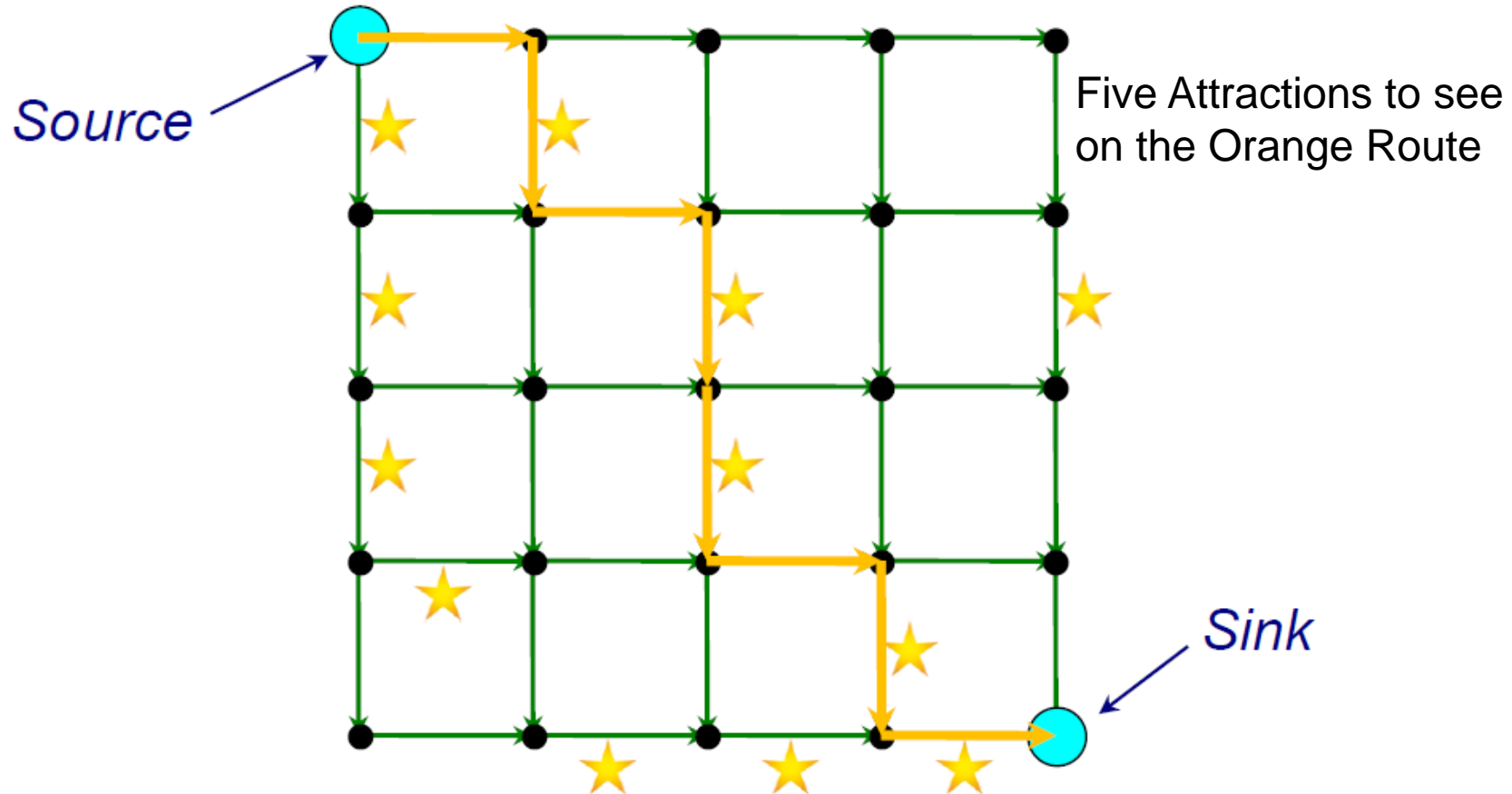
A sequence alignment diagram showing two DNA sequences. The top sequence is A C G T G C G C T G - C T G -. The bottom sequence is - C G T - C - C T G C C T G C. Vertical arrows connect the characters: red arrows indicate mismatches (A to -, G to C, G to -, G to C, G to C, T to G, T to -, G to C) and green arrows indicate matches (C to C, T to T, C to C, C to C, T to T, G to G, C to C, T to T, G to G).

A C G T G C G C T G - C T G -
- C G T - C - C T G C C T G C

- This one seems to be better. How can we find optimal alignments?

First, an Analogy for sequence alignment

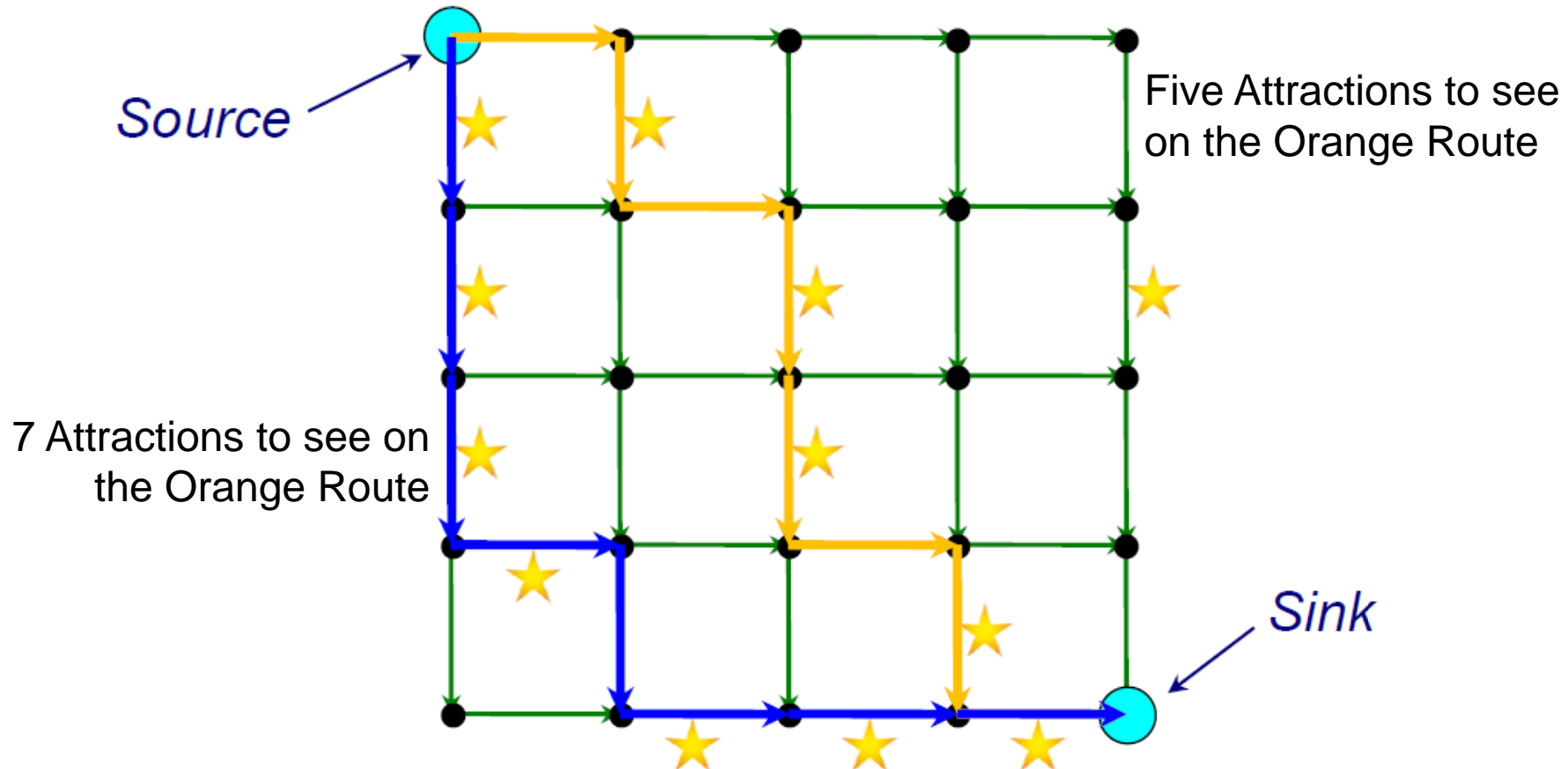
- The Manhattan Tourist Problem:



- A tourist on the upper west side, walking to the lower east side, wants to see as many sights as possible. The tourist walks only east and south.

First, an Analogy for sequence alignment

- The Manhattan Tourist Problem:



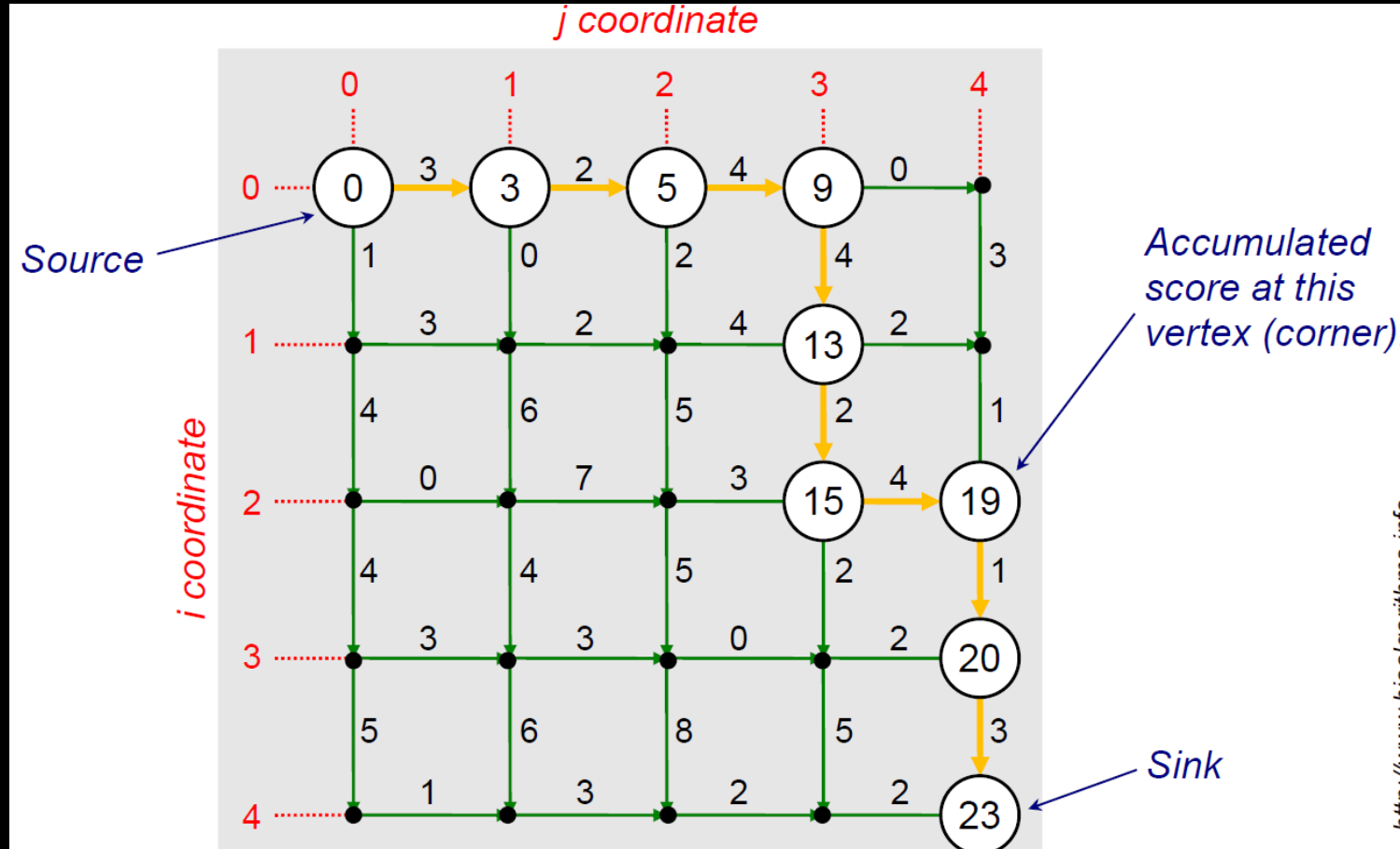
- A tourist on the upper west side, walking to the lower east side, wants to see as many sights as possible. The tourist walks only east and south.

Formality: The Manhattan Tourist Problem

- Find the highest value path on the grid
- Input: A weighted directional graph G with two distinct vertices, one labeled “source” and the other labeled “sink.”
- The weights are predetermined based on tourist preference
- Output: The path in G from “source” to “sink” with highest score

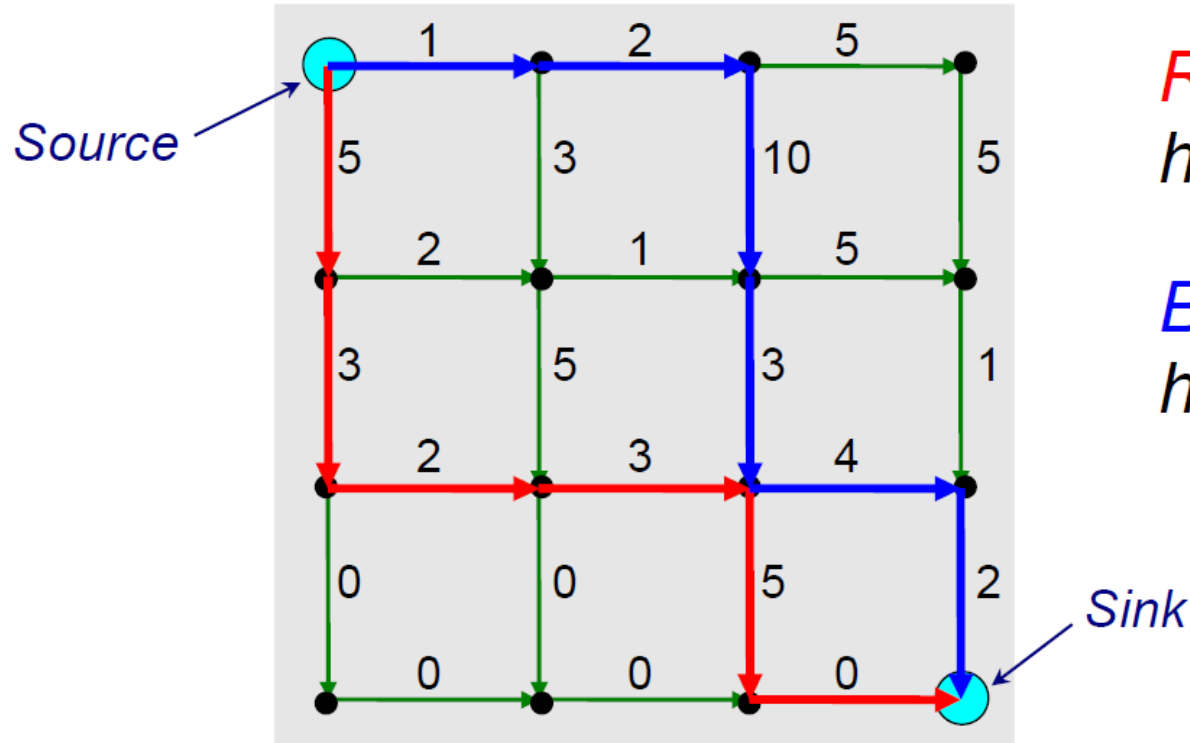
We can do this in a Greedy way

- Find the best outgoing edge at every corner



- Start in the upper left, and just pick the higher scoring alternative

Greediness doesn't always pay



- Here, the greedy approach scores only 18, whereas the optimal path is 22.

A recursive approach to the MTP

MT(n, m):

If ($n = 0$) **and** ($m = 0$):

Return 0;

If ($n > 0$):

$x = \text{MT}(n-1, m) + \text{length of edge from } (n-1, m) \text{ to } (n, m)$

else

$x = 0$;

If ($m > 0$)

$y = \text{MT}(n, m-1) + \text{length of edge from } (n, m-1) \text{ to } (n, m)$

else

$y = 0$;

Return $\max(x, y)$;

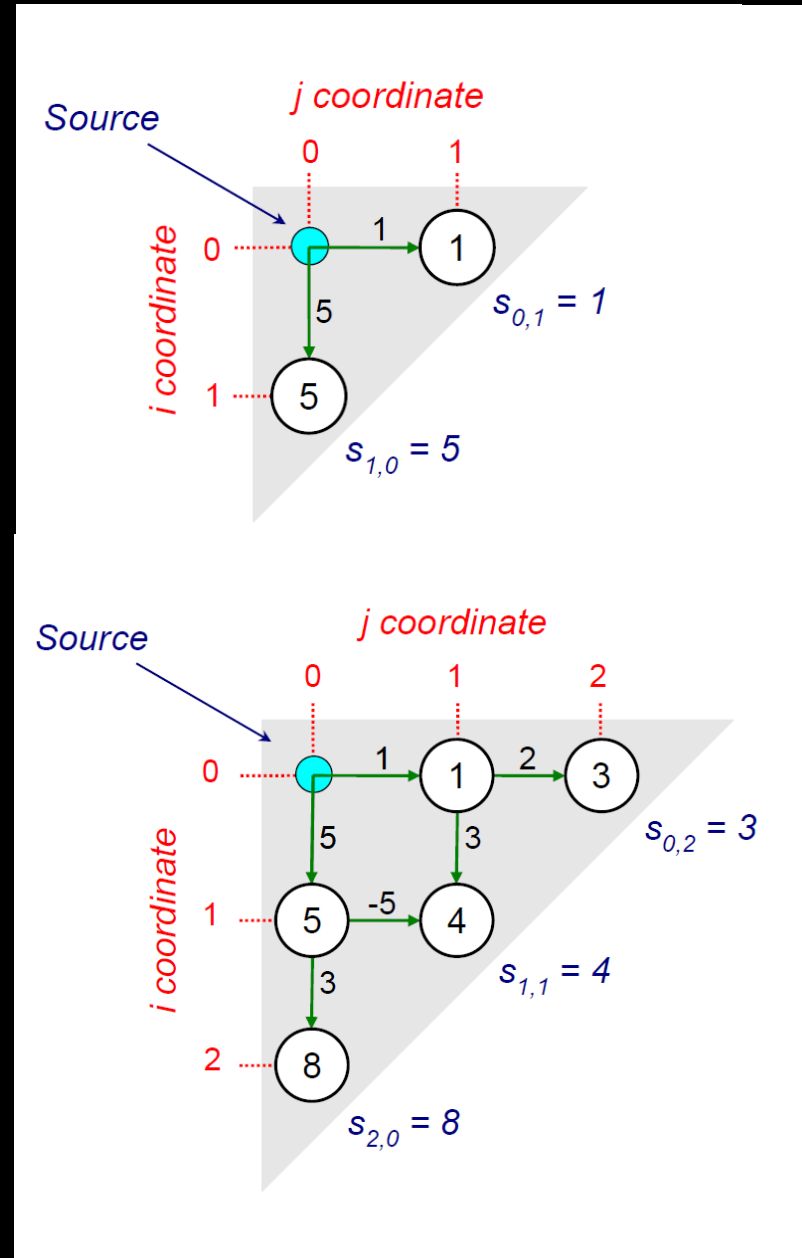
This algorithm captures an important idea:

The value of later square can be
Dependent on earlier squares

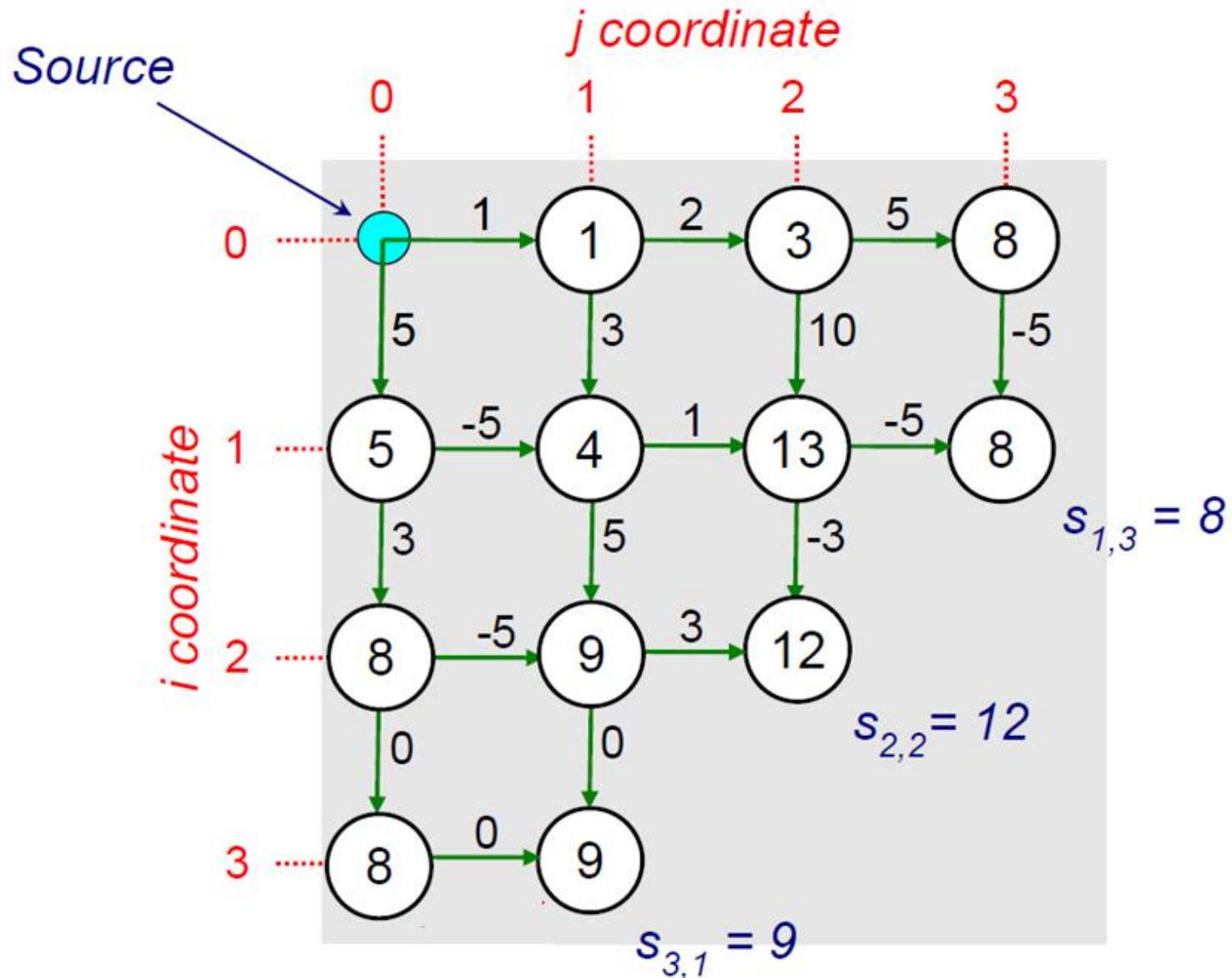
Unfortunately this recursive algorithm
would be very very slow!

A better way to evaluate these scores

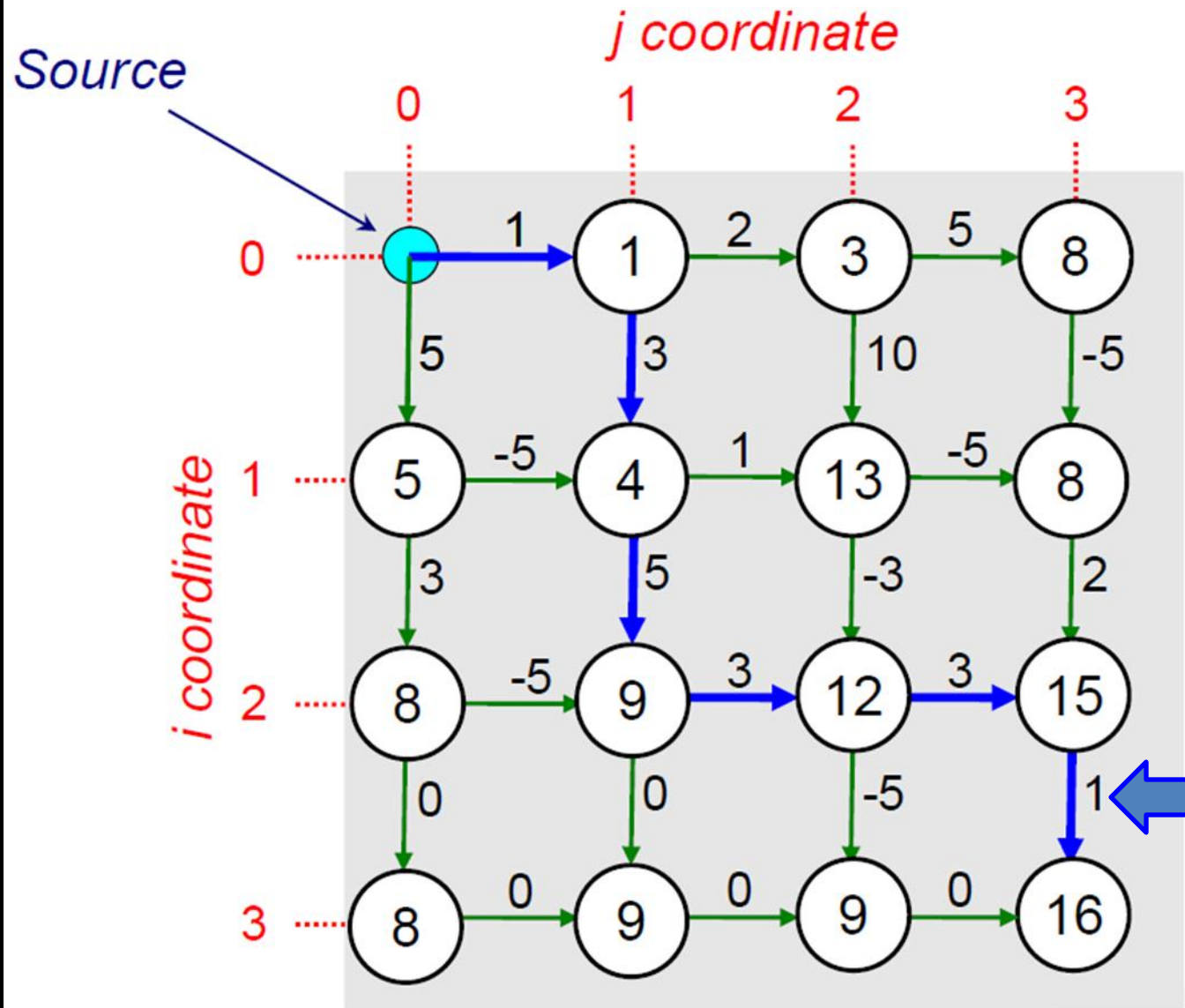
- Subsequent scores for recursive MTP are simply the $\max()$ of previous scores
- We can compute everything once, without constantly going back to recompute.



Here we compute two more steps



Now we've completed the calculation



Now we can step backwards to the highest scoring node to find the highest scoring path

A formal statement of node score

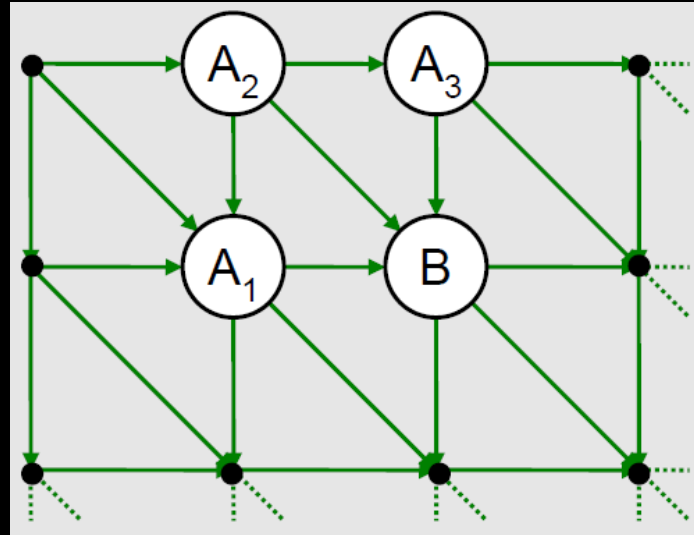
- We compute the score for each node (i,j) based on the following recurrence relation:

$$s_{i,j} = \max \begin{cases} s_{i-1,j} + \text{weight of the edge between } (i-1, j) \text{ and } (i, j) \\ s_{i,j-1} + \text{weight of the edge between } (i, j-1) \text{ and } (i, j) \end{cases}$$

- The running time for this approach is $n*m$, for n rows and m columns

But Manhattan has diagonal streets

- We can adapt the Manhattan Tourist Problem for maps with diagonal streets.



The score
at node B

$s_B =$

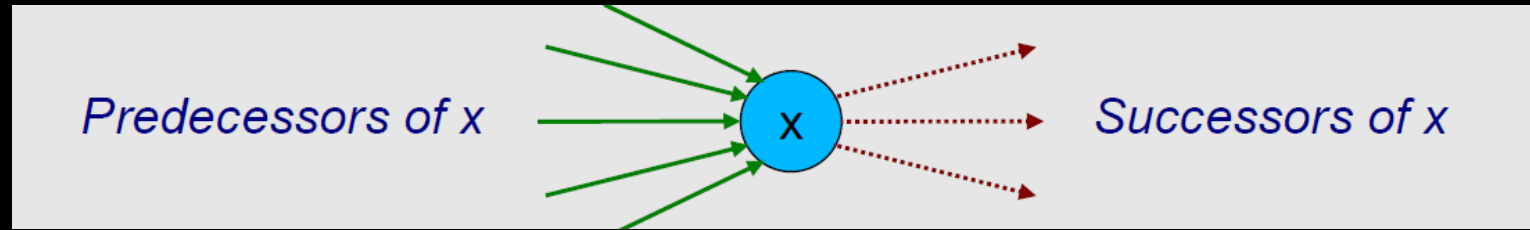
max
of

$s_{A_1} + \text{weight of the edge } (A_1, B)$

$s_{A_2} + \text{weight of the edge } (A_2, B)$

$s_{A_3} + \text{weight of the edge } (A_3, B)$

In fact, a total generalization of the method:



- A directed graph has only predecessors and successors of a given node x .
- We can thus compute the score of the node x :

$$s_x = \max_{\text{of}} \left\{ s_y + \text{weight of edge } (y, x) \text{ where } y \in \text{Predecessors}(x) \right.$$

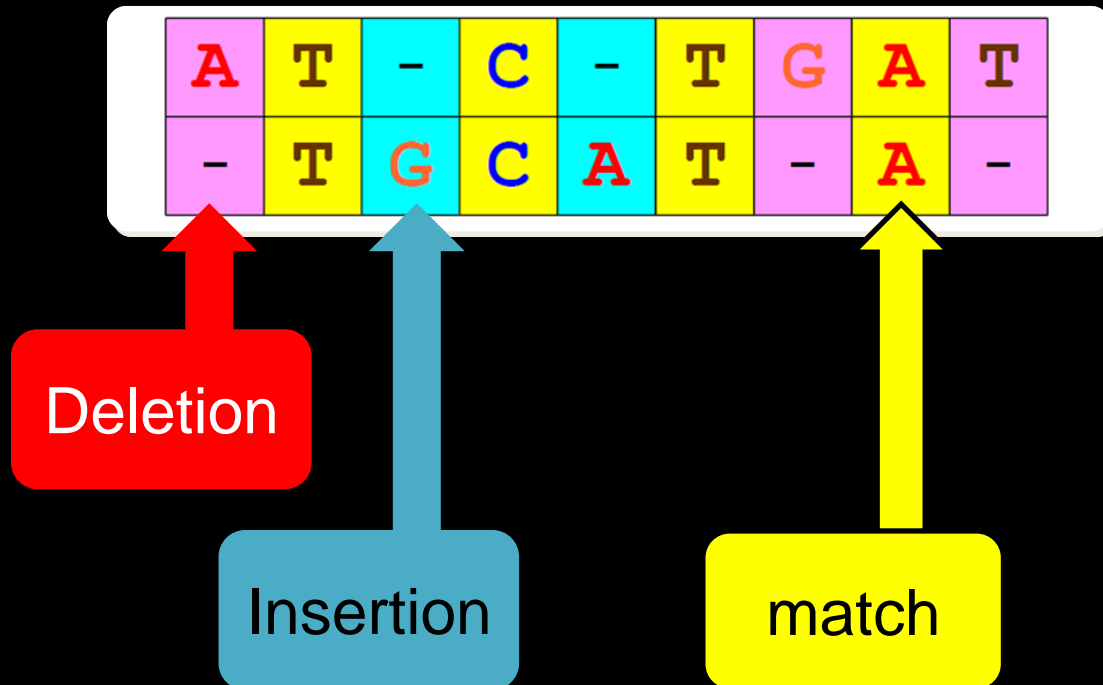
- Here the runtime of the method for a graph with E edges will be $O(E)$, since each edge is considered once.

What this all has to do with sequences

- Suppose we have two DNA sequences, v , w :

$v =$ **A** **T** **C** **T** **G** **A** **T** $m = 7$
 $w =$ **T** **G** **C** **A** **T** **A** $n = 6$

- An alignment of v and w looks like this:



The Longest Common Subsequence problem

Find the longest common subsequence (LCS) of two sequences.

Input: Two sequences:

$$V = V_1 V_2 \dots V_m \qquad W = W_1 W_2 \dots W_n$$

Output:

A series of positions in v :

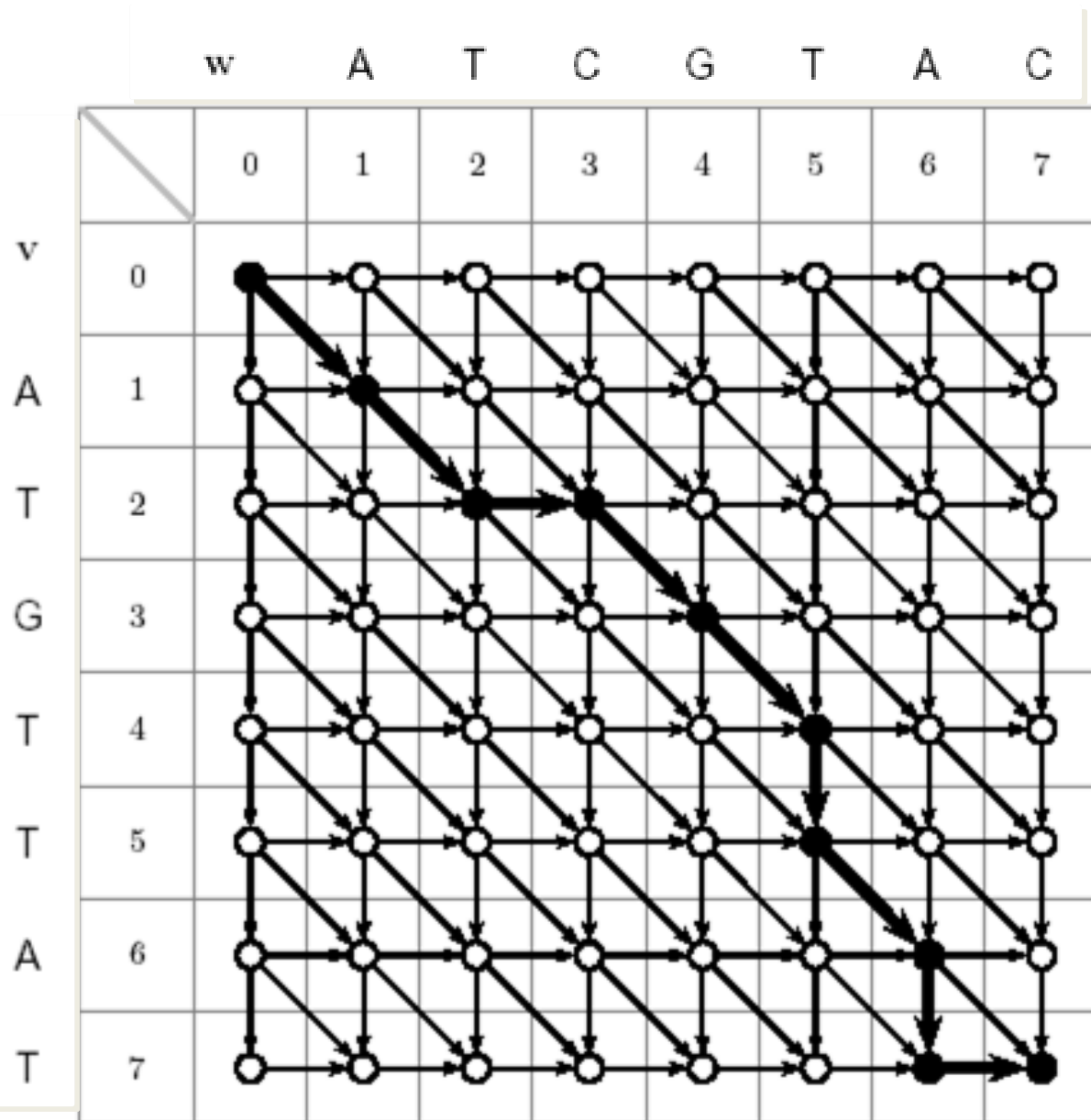
$$1 \leq i_1 < i_2 < \dots < i_t \leq m$$

and a series of positions in w :

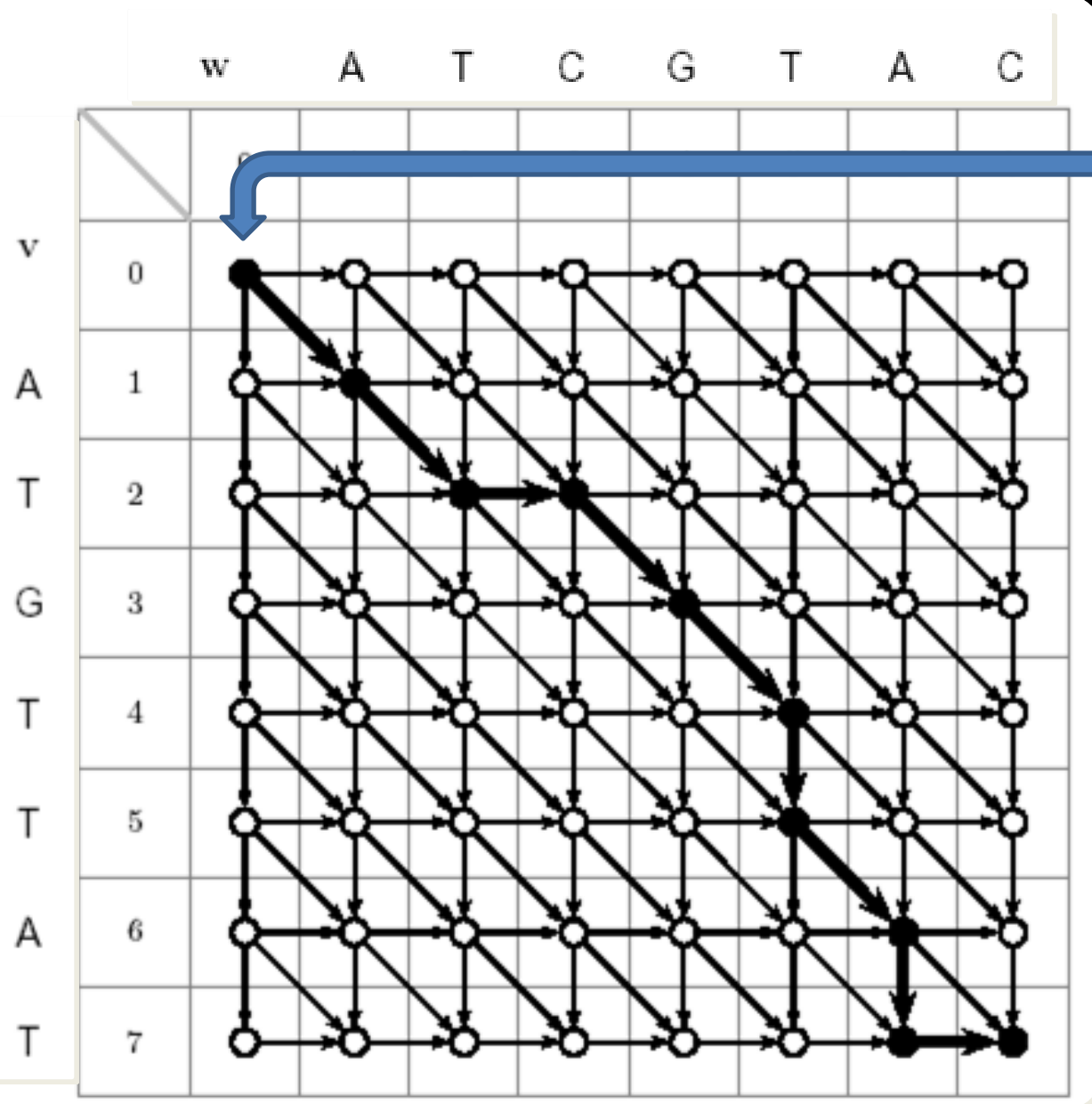
$$1 \leq j_1 < j_2 < \dots < j_t \leq n$$

such that symbol i_k of v matches j_k of w and t is maximal

An MTP graph can compute the LCS

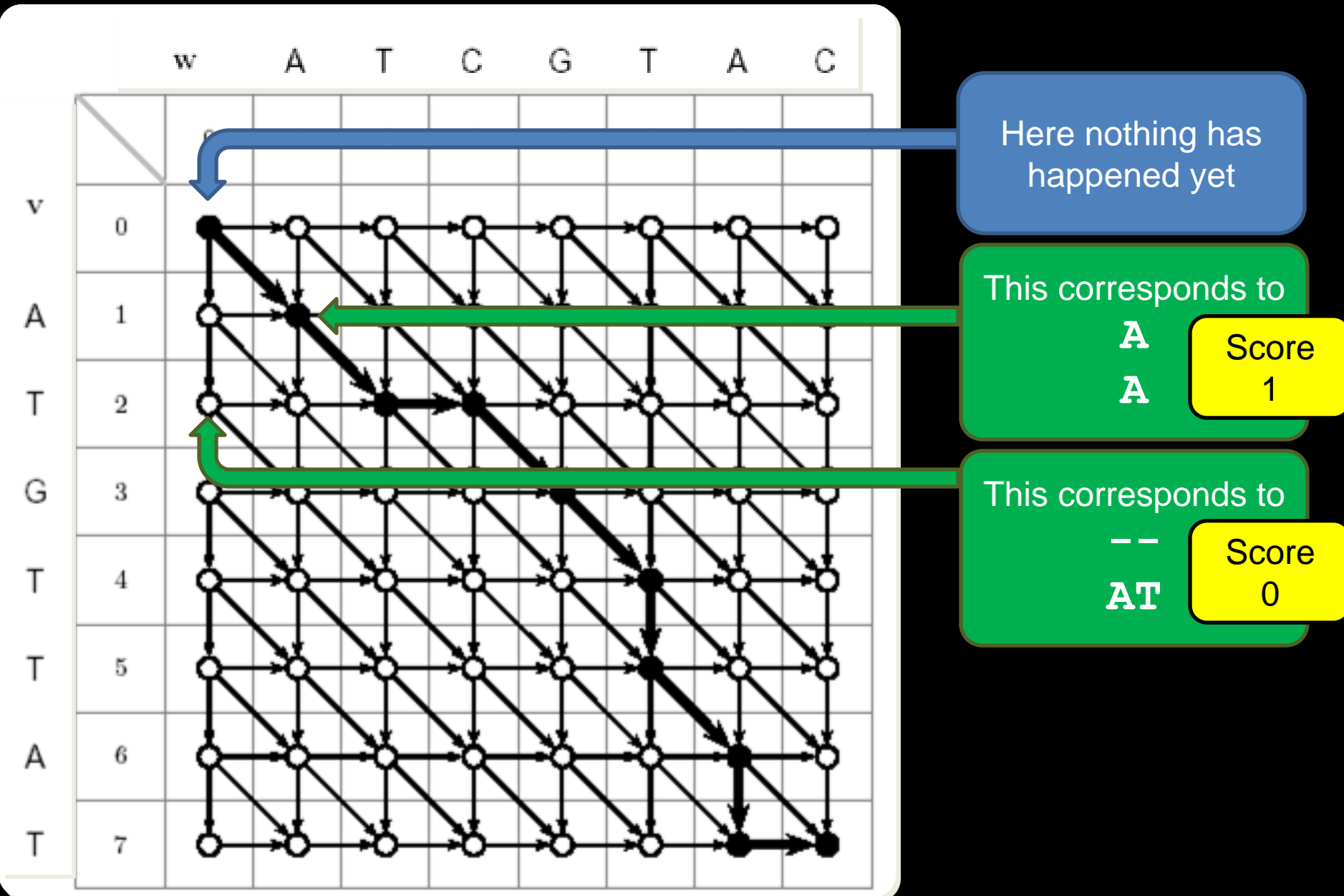


An MTP graph can compute the LCS

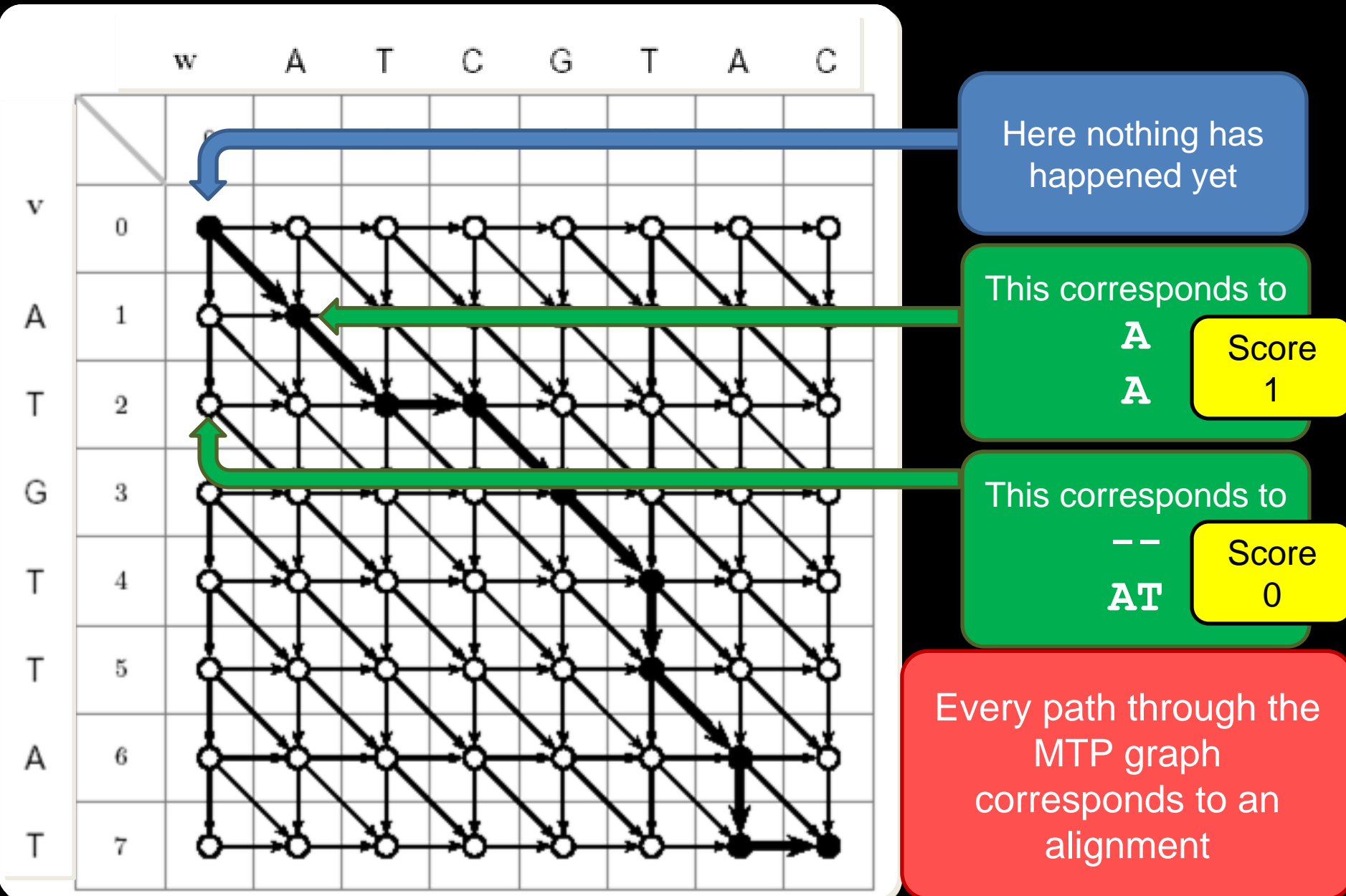


Here nothing has happened yet

An MTP graph can compute the LCS

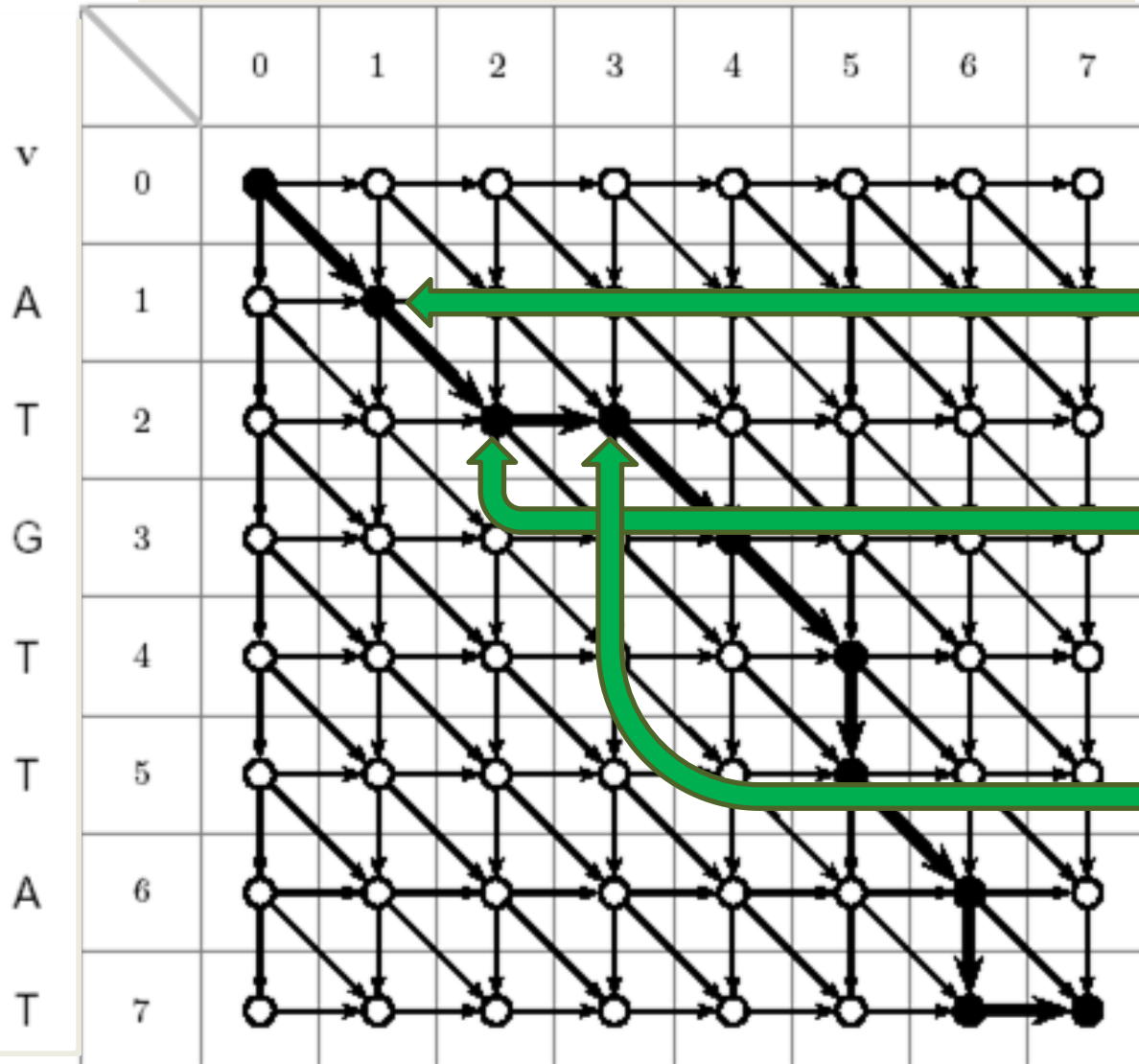


An MTP graph can compute the LCS



Following the bolded path..

Every path through the MTP graph corresponds to an alignment



This corresponds to

A
A

Score
1

This corresponds to

AT
AT

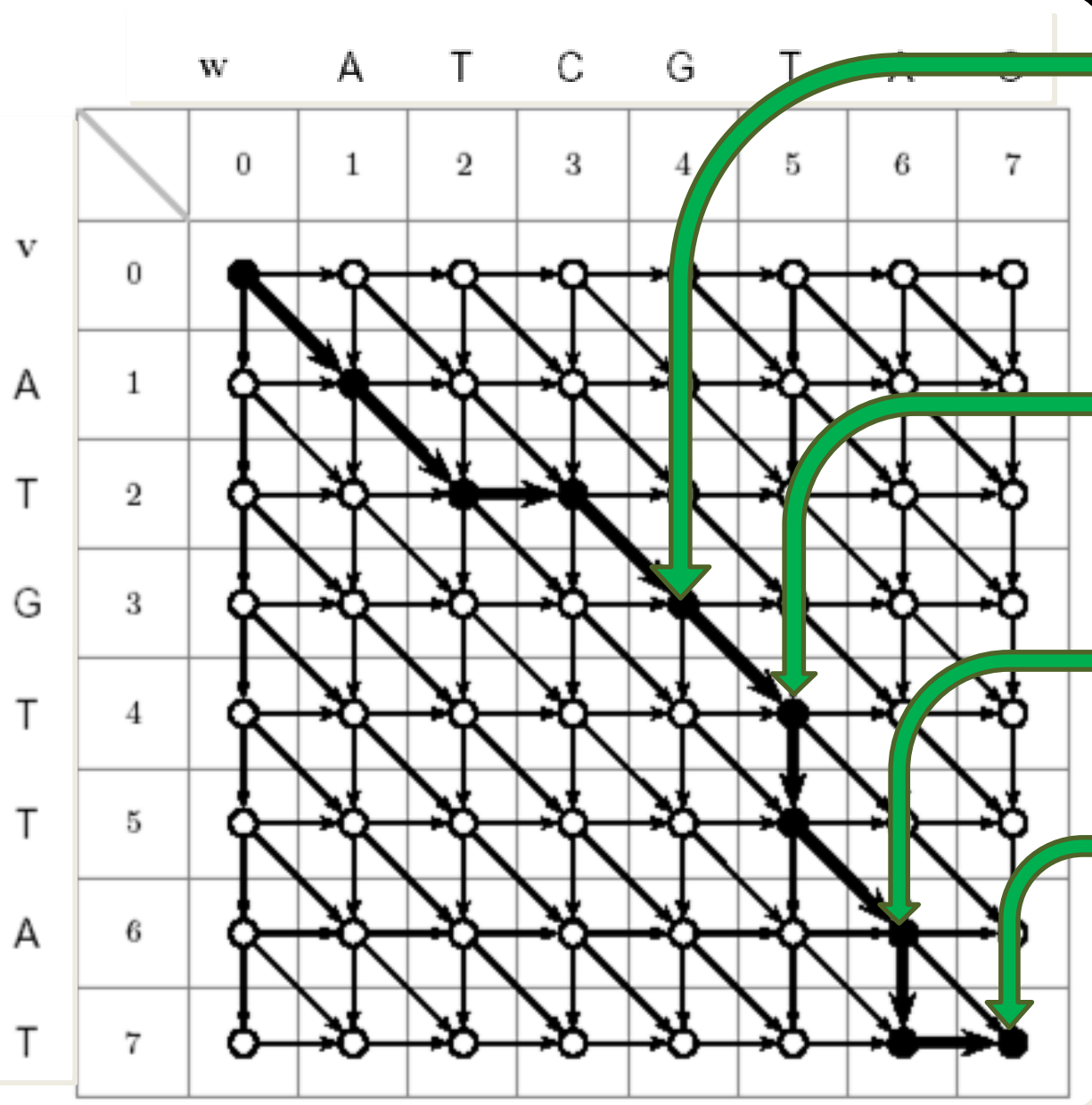
Score
2

This corresponds to

ATC
AT-

Score
2

Following the bolded path..



This corresponds to

ATCG

AT-G

Score

3

This corresponds to

ATCGT

AT-GT

Score

4

This corresponds to

ATCGT-A

AT-GTTA

Score

4

This corresponds to

ATCGT-A-C

AT-GTTAT-

Score

4

How are we populating the scores?

- Nucleotide substitution scores are based on hand curated datasets of protein sequences
 - Yes, people actually aligned things by hand to make sure they were getting the right alignments
- Based on many hand curated sequences, it is possible to measure the frequency of substitution for different nucleotides

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	1	-4	-4	-4
C	-1	-4	1	-4	-4
G	-1	-4	-4	1	-4
T	-1	-4	-4	-4	1

Source:
<http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/new/node90.html>

Summary

- We have seen how the problem of sequence alignment corresponds to the dynamic programming Manhattan Tourist Problem
- Each path through the graph corresponds to a sequence alignment
- We can score the alignment based on nucleotide substitution rules observed elsewhere
- After we have scored the matrix, we must walk back through to get the optimal path: we will see how to do that next time.

Questions